Mobility Analytics Based on Passive Sensing Data

Dissertation

zur Erlangung des Grades eines

D o k t o r s d e r I n g e n i e u r w i s s e n s c h a f t e n

der Technischen Universität Dortmund
an der Fakultät für Informatik

von

Yunfeng Huang

Dortmund

2025

Tag der mündlichen Prüfung:	22.05.2025
Dekan:	Prof. Dr. Peter Buchholz
Gutachter:	Prof. Dr. Jian-Jia Chen
	Prof. Dr. Fang-Jing Wu

Abstract

Nowadays, mobility analytics plays an important role in our daily lives, such as in improving urban planning, optimizing transportation systems, and enhancing public safety. With the rapid development of sensor technology, contemporary mobility analytics based on sensing data has become more efficient and reliable. Depending on how the sensing data is acquired, mobility analytics is generally divided into active sensing-based mobility analytics and passive sensing-based mobility analytics.

Active sensing-based mobility analytics, such as deploying Light Detection and Ranging (LiDAR) sensors for traffic flow monitoring or Radio Detection And Ranging (RADAR) systems for movement tracking, either requires a high cost of deploying additional infrastructure or users' active and continuous participation in sensing data collection. In addition, privacy concerns may arise due to the exposure of personal identity in active sensing technologies.

Passive sensing-based mobility analytics, such as Wi-Fi-based localization or Inertial Measurement Unit (IMU)-based navigation, passively collects data from existing Internet of Things (IoT) sensors in the environment. Because these IoT sensors are not dedicated to mobility analytics but pre-exist to support other IoT services, passive sensing-based mobility analytics avoids the high costs of installing additional infrastructure, the need of users' active participation, and the exposure of users' identities.

However, passive sensing-based mobility analytics still faces many limitations. First, since passive sensing technology lacks dedicated IoT infrastructure, sensing data from a single sensor usually provides mobility information from a limited perspective. Second, the process of collecting passive sensing data is uncontrollable because ambient sensors are not directly controlled by users. As a result, the potential data loss may cause failures in mobility analytics when one type of sensor stops providing data for unknown reasons. Third, passive sensing data is generally sparse also due to this uncontrollable data collection process. Therefore, many studies aim to generate denser sensing data to enhance mobility analytics. However, generating reliable sensing data is non-trivial and remains challenging.

To address these limitations, this dissertation proposes a **collaborative** and **complementary** computing paradigm for passive sensing-based mobility analytics. The key idea behind the proposed paradigm lies in three aspects: 1) mobility analytics based on multi-modal sensing data, 2) mobility analytics based on cross-domain sensing data, 3) multi-model-based sensing data generation. In the first case, different types of sensors are jointly utilized for mobility analytics to complement the weaknesses of individual sensors. In the second case, different forms of sensing data from the same sensor are incorporated to provide insights from different knowledge domains. In the third case, Artificial Intelligence (AI)-driven methods and non-AI-driven methods are synergized to generate denser sensing data.

In this dissertation, the feasibility of the proposed computing paradigm is first demonstrated in our preliminary work, where the advantages of collaboration and complement between different sensors are exhibited. Next, this dissertation further investigates the necessity and effectiveness of the proposed computing paradigm through comprehensive mobility analytics in the following three scenarios.

- This dissertation estimates **Physical Proximity** between users based on multi-modal sensing data, i.e., Wi-Fi data and IMU data. Wi-Fi data provides absolute spatial information for mobility analytics, which IMU data lacks. Conversely, IMU data offers fine-grained mobility information, which is not available in Wi-Fi data. The joint use of Wi-Fi data and IMU data complements each other's weaknesses, facilitating more reliable physical proximity estimation.
- This dissertation investigates users' **Visual Attention** based on cross-domain sensing data, i.e., eye movements and light patterns reflected in human eyes. The movement of human eyeballs and the light patterns reflected in human eyes characterize users' visual attention from different knowledge domains. Therefore, the joint use of both types of data enables a more comprehensive analysis of users' visual attention.
- This dissertation develops a framework for **Indoor Localization** through multi-model-based Wi-Fi fingerprint generation. First, Wi-Fi radio maps are augmented by jointly utilizing a Generative Adversarial Network (GAN) model and a Gaussian Process Regression (GPR) model,leveraging the strengths of each approach. Second, a tailored localization algorithm is designed by incorporating the augmented Wi-Fi radio maps.

This dissertation provides a comprehensive and in-depth mobility analytics based on the proposed paradigm. On the one hand, the importance of collaboration and complementarity in passive sensing-based mobility analytics is validated. On the other hand, feasible strategies for mobility analytics in different scenarios are given in this dissertation.

List of Publications

Part of the findings and corresponding methodologies in this dissertation has been published in international journals and conferences ([WHD+20; HWH+20; HW21]), and another part is being prepared for journal submission ([HW25b; HW25a]).

[WHD+20] F.-J. Wu et al. "PassengerFlows: A correlation-based passenger estimator in automated public transport". In: *IEEE Transactions on Network Science and Engineering* 7.4 (2020), pp. 2167–2181

[HWH+20] Y. Huang et al. "Demo Abstract: Perception vs. Reality-Never Believe in What You See". In: *2020 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE. 2020, pp. 363–364

[HW21] Y. Huang and F.-J. Wu. "CRISIS: Cyber-physical social distancing based on multi-modal data from mobile devices". In: *IEEE Transactions on Mobile Computing* 22.5 (2021), pp. 2551–2568

[HW25b] Y. Huang and F.-J. Wu. "V-Groups: Matching Light Traces on Human Eyes for Detecting Visual Attention Groups". In: *(Preparing for submission)* (2025)

[HW25a] Y. Huang and F.-J. Wu. "Spatial Reconstruction and Localization based on Radio Map Adaption to Time-varying Environments". In: *(Preparing for submission)* (2025)

[HEW+22] Y. Huang et al. "Half-Farmer: A Human-Machine Augmented Learning Framework for Seed Germination Recognition in Smart Farming". In: *2022 IEEE 8th World Forum on Internet of Things (WF-IoT)*. IEEE. 2022, pp. 1–6

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my supervisor, Prof. Fang-Jing Wu, for her invaluable support during my PhD journey. She has helped me far beyond what one would expect from a supervisor, not only by providing insightful advice on research, but also by offering encouragement and patience whenever I encountered difficulties or challenges. I am fully aware of my own limitations and imperfections. Nevertheless, she has consistently placed her trust in me and stood by me during challenging periods. To me, she has been not only a supervisor but also a trusted mentor and friend, whose influence has extended far beyond the realm of research. Under her guidance, I gradually learned to truly appreciate the value and joy of research.

I would also like to sincerely thank Prof. Jian-Jia Chen for his continuous support throughout my doctoral studies. Without his encouragement during the early stages of my PhD, as well as his tremendous support during the final and most critical phase after my supervisor left TU Dortmund University, I would not have been able to successfully complete my PhD.

Furthermore, I would like to thank Prof. Peter Buchholz and Prof. Christian Janiesch for taking the time to review my dissertation and for participating in my PhD defense. I sincerely appreciate their thoughtful comments and constructive suggestions, which have helped improve the quality of this dissertation.

Finally, I would like to thank my parents for their unconditional love, understanding, and support throughout the years. Their encouragement has always been a constant source of strength for me. I would also like to remember my grandmother, who passed away during the time I was writing this dissertation and was unable to witness the completion of this important milestone. Her love and support will always remain in my heart and continue to inspire me. My deepest gratitude goes to my wife. Throughout this journey, her companionship, patience, and enduring care have been essential to me. During the most difficult and uncertain periods of my PhD, she stood by me with understanding and encouragement, offering emotional strength and stability when I needed them most. Her constant presence, trust, and quiet support have sustained me far beyond my academic life, and without her, I could not have completed this journey.

Contents

List of Publications iii

1 Introduction 1
- 1.1 Background and Context . . . 1
- 1.2 Motivation . . . 5
- 1.3 Preliminary Work . . . 8
- 1.4 Goals, Challenges, and Contributions . . . 11
- 1.5 Author's Contribution to this Dissertation . . . 20

2 Related Work 21

3 Mobility Trajectory Comparison Based on Multi-Modal Sensing Data 27
- 3.1 System Model . . . 27
  - 3.1.1 System Overview . . . 28
- 3.2 Creation of Multi-modal Signatures . . . 29
  - 3.2.1 Creation of Mobility Signatures . . . 29
  - 3.2.2 Creation of Movement Signatures . . . 30
- 3.3 Quantification of Cyber Distances . . . 30
  - 3.3.1 Inter-similarity between Mobility Signatures . . . 32
  - 3.3.2 Self-Similarity of Mobility Signatures . . . 35
  - 3.3.3 Similarity between Movement Signatures . . . 35
- 3.4 Cyber-physical Social Distancing . . . 36
  - 3.4.1 Detection Algorithm . . . 36
  - 3.4.2 Complexity of the Detection Algorithm . . . 36
- 3.5 Implementation and Demonstration . . . 38
- 3.6 Experiments and Evaluation . . . 38
  - 3.6.1 Small-scale Environments and Mobility Scenarios . . . 39
  - 3.6.2 Performance Metrics . . . 41
  - 3.6.3 Effects of Mobility Trajectories . . . 42
  - 3.6.4 Analyses of Device-to-Device Cyber Distances . . . 42
  - 3.6.5 Advanced Study in a Large-Scale Environment . . . 46
  - 3.6.6 Synchronicity between Cyber and Physical distances . . . 58
  - 3.6.7 Analyses of Execution Time . . . 61
- 3.7 Conclusion . . . 63

4 Visual Trajectory Comparison Based on Cross-Domain Sensing Data **65**
4.1 System Model . . . 65
4.1.1 Problem Statement . . . 65
4.1.2 System Framework . . . 65
4.2 Creation of Visual Signatures . . . 66
4.2.1 Workflow of Creating Visual Signatures . . . 67
4.2.2 Similarity Score between Visual Signatures . . . 68
4.3 Detection of Visual Transitions . . . 70
4.3.1 Computation of Attention Level . . . 71
4.3.2 Detection of Visual Transitions . . . 72
4.4 Quantification of Visual Attention Similarity . . . 73
4.4.1 Computation of Pairwise Similarity Scores . . . 73
4.4.2 Computation of Visual Attention Similarity . . . 75
4.5 Implementation . . . 77
4.6 Experiments and Evaluation . . . 78
4.6.1 Preliminary Analysis based on Controlled Experiments . 79
4.6.2 Advanced Analysis based on Uncontrolled Experiments . 87
4.7 Conclusion . . . 91

5 Multi-Model-Based Generation of Sensing Data **93**
5.1 System Model . . . 93
5.1.1 System Modelling . . . 93
5.1.2 System Overview . . . 94
5.2 Adversarial Learning for Radio Map Generation . . . 95
5.2.1 Framework Design . . . 95
5.2.2 Network Architecture . . . 96
5.3 RSS Correlation-based Fingerprinting and Localization . . . 97
5.3.1 Creation of Fingerprints . . . 97
5.3.2 Location Predictor . . . 100
5.4 Experiments and Evaluation . . . 100
5.4.1 Experimental Setup . . . 101
5.4.2 Network Implementation . . . 102
5.4.3 Comprehensive Analysis in a Large-Scale Environment . . 104
5.4.4 Advanced Study in various Environments . . . 109
5.5 Conclusion . . . 115

6 Conclusions, Discussion, and Outlook **117**
6.1 Summary . . . 117
6.2 Discussion . . . 119
6.3 Future Work . . . 121

List of Figures **124**

List of Tables 125

List of Algorithms 127

Glossary 129

Bibliography 131

Chapter 1

Introduction

1.1 Background and Context

The rapid development of sensing technology has created more opportunities to interact with the physical world from different perspectives. By measuring and detecting chemical, physical, and biological phenomena, various sensors convert observations of the world into sensing data that can be processed by computer systems. Specifically, such sensing data can convey environmental information (e.g., temperature and humidity), spatial information (e.g., velocity and orientation), biological information (e.g., heart rate and respiration rate), and behavioral information (e.g., gesture patterns and movement trajectories). Consequently, a wide range of services across diverse application domains can be supported more efficiently such as autonomous driving, intelligent transportation, smart manufacturing, healthcare monitoring, and next-generation communication systems [KRA+20].

In recent years, the growth of urban populations and the increasing demand for transportation services have made traffic systems more congested, dynamic, and difficult to manage. As Internet of Things (IoT) infrastructure equipped with diverse sensors becomes increasingly prevalent, sensing-based mobility analytics has emerged as a critical tool for understanding and optimizing the movement of people, goods, and vehicles, thus contributing to smarter urban services and improved quality of daily life. For example, the works in [SMB+17; SWX+20; MSB+21] improve urban planning by analyzing human mobility during travel, providing reliable predictions of travelers' behaviors and preferences. The works in [RHZ+21; WLQ+20; ZLL+20] optimize public transportation systems by investigating the mobility of both crowds and vehicles, effectively recognizing movement patterns under different traffic conditions. The works in [CLZ+22; ZTC+22; PCM+22] enhance public safety by estimating human-to-human proximity in public spaces, enabling timely alerts that help maintain appropriate social distance and reduce the risk of disease transmission. Furthermore, mobility analytics also plays an important role in many IoT-based services, such as estimating users' locations in shopping malls to support proximity-based advertisements [BYC+21], monitoring elderly mobility to detect falls and prevent injuries [JSC20], and improving logistics by better protecting temperature-sensitive goods [YWW+21].

Sensing-based mobility analytics is generally categorized into two types according to how sensing data is collected: 1) mobility analytics based on active sensing technology and 2) mobility analytics based on passive sensing technology, as shown

in Fig. 1.1. Active sensing-based approaches rely on dedicated devices that actively generate sensing signals and often consume extensive energy to monitor the mobility of targeted entities. In contrast, passive sensing-based approaches require neither additional dedicated devices nor active user participation. Instead, they leverage existing infrastructures and devices, originally deployed for other purposes, to perform mobility analytics.

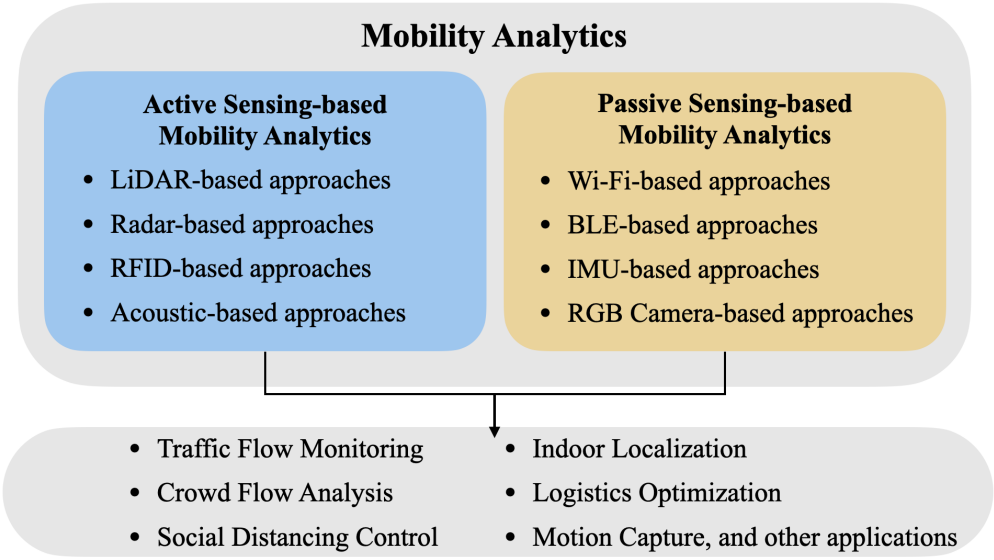


Figure 1.1: Mobility analytics based on sensing technology.

Active Sensing-based Mobility Analytics

Traditional mobility analytics largely relies on active sensing technology, where "active" refers to either the active transmission of sensing data by the source or the active request for sensing data by the receiver. In active sensing technology, sensing data is typically generated by dedicated sensors that are pre-deployed in the environment, such as Light Detection and Ranging (LiDAR), Radio Detection And Ranging (RADAR), and ultrasonic sensors. These active sensors consume a substantial amount of energy to produce sensing data. Specifically, they first probe the ambient environment using various types of waves, such as radio waves, laser pulses, or ultrasound waves, and then convert their environmental perceptions into digital signals that can be processed by computer systems for mobility analytics. In addition, users also actively interact with dedicated sensors in active sensing technology, thus participating in the collection of sensing data. For example, in mobility analytics based on Radio Frequency Identification (RFID) sensors, users wearing RFID tags interact with RFID readers deployed in the environment, thereby generating sensing data that reflects their mobility patterns.

Studies such as [ZXC+20; CXW+19] deploy LiDAR sensors along roadways to detect and track vehicles, providing reliable estimates of vehicle speeds. Research in [ZXL+19; ZLX+19; TBB+23] leverages pre-installed LiDAR sensors for pedestrian detection to improve traffic monitoring and support urban planning. The works in [LCW+23; GA19] utilize RADAR sensors for indoor mobility analytics, where [LCW+23] estimates user locations and [GA19] develops a deep-learning method for motion recognition. Radar sensors are also applied to indoor occupancy detection, as demonstrated in [CKK21], providing potential solutions for mitigating disease transmission during pandemics. Furthermore, studies such as [BXG+20; LHY+18; WLC+18] deploy large-scale RFID tags in the environment for human motion recognition, supporting various IoT-based applications including fall detection, human-computer interaction, and industrial safety and surveillance.

Active sensing-based mobility analytics can provide accurate and stable mobility information, as dedicated devices offer high-quality sensing data and, in some systems, user interaction ensures data consistency. Thus, active sensing remains valuable in specific application scenarios today. However, it also faces notable limitations. First, active sensing technology often involves high deployment and maintenance costs due to the use of expensive sensors or large-scale sensor networks, which significantly restricts its scalability. Second, many active sensing systems require users to participate in data collection, yet user involvement cannot always be guaranteed in reality. Finally, active sensing systems frequently depend on user identification, raising privacy concerns that may discourage participation from privacy-conscious individuals.

Passive Sensing-based Mobility Analytics

Over the past few years, the explosive growth of IoT-based services has led to the widespread deployment of IoT sensors in many aspects of daily life, including Wi-Fi Access Points (APs), Bluetooth Low Energy (BLE) beacons, Inertial Measurement Unit (IMU) sensors on smartphones, and different built-in sensors in wearable devices. Therefore, leveraging these existing sensors for mobility analytics, also referred to as passive sensing-based mobility analytics, has attracted increasing attention from both industry and academia. From an industry perspective, commercial applications built upon existing sensors can significantly reduce the costs associated with deploying dedicated sensing infrastructure, thereby enhancing market competitiveness. From a research perspective, it is valuable to explore new models and frameworks that fully capitalize on existing sensors to provide reliable and scalable mobility analytics.

Supported by various sensors widely available in the environment, mobility analytics based on passive sensing technology has penetrated into all aspects of our lives. The works in [YYT22; ABD+23; ZWZ23] estimate users' locations in the indoor environment based on Wi-Fi signals. These works capture the Received Signal Strength (RSS) values from different Wi-Fi APs in the ambient environment, which

implies the physical distances between users and the Wi-Fi APs based on the nature of wireless signal propagation. The works in [GLL+23; GJ23; YZZ22] also investigate users' mobility based on the information from Wi-Fi APs, where [GLL+23] recognizes users' gestures for human-computer interaction, [GJ23] tracks users' motions by analyzing the variations in Wi-Fi signals caused by users' movements, and [GJ23] devises a deep learning system for fall detection based on Wi-Fi signals. The works in [ZZH+22; LJW+21; DTC+21; MS23] analyze users' mobility using the signals from existing BLE beacons, where users' locations are predicted in [ZZH+22], the distance between moving and static users are estimated in [LJW+21], and an indoor occupancy detection system is designed in [DTC+21]. As IMU sensors are commonly embedded in various devices such as smartphones, smart glasses or other smart wearable devices, IMU data can be also passively collected for mobility analytics. Based on the collected IMU data, the work in [GRN+24] analyzes users' gait for fall detection, the work in [ZLL+24] investigates thoracic spine mobility for disease monitoring, and the work in [YZX21] captures users' motion for body poses estimation.

Passive sensing technology offers several important advantages over active sensing for mobility analytics, as summarized in Table 1.1. First, passive sensing-based mobility analytics does not require dedicated sensing devices, which greatly reduces deployment cost and improves system scalability. Second, users do not need to actively interact with sensing sources or repeatedly request data from sensors. This ease of use significantly improves deployability and makes passive sensing approaches broadly applicable in real-world scenarios. Third, active sensing typically relies on energy-intensive sensors such as LiDAR or RADAR, whereas passive sensing data can be obtained at very low energy overhead from ambient sources such as Wi-Fi APs or BLE beacons. Finally, passive sensing-based mobility analytics generally relies on physical-layer signal measurements, such as RSS, Channel state information (CSI), Angle of Arrival (AoA), and Time of Arrival (ToA), rather than on user identity information. Note that, when identity information and sensing data are collected concurrently, encryption techniques can be applied to anonymize identifiers, thereby providing an additional layer of privacy protection. Consequently, users with privacy concerns can also access the corresponding services with confidence.

Table 1.1: Advantages of passive sensing-based mobility analytics.

	Active sensing-based mobility analytics	**Passive sensing-based mobility analytics**
High scalability		✓
No active user participation		✓
Low energy consumption		✓
Privacy preservation		✓

Although passive sensing-based mobility analytics demonstrates many advantages, it still has some limitations:

(1) **Unreliable analytics based on single-modal sensing data.** Passive sensing-based mobility analytics relies on existing IoT infrastructures rather than dedicated sensing devices. However, because these infrastructures are originally deployed for other purposes, they cannot consistently provide reliable sensing support. In particular, when certain IoT devices become unavailable due to removal, malfunction, or temporary signal loss, the mobility analytics that depends on them can be substantially degraded. Therefore, mobility analytics that relies exclusively on single-modal sensing data from a single type of sensor is unreliable.

(2) **Limited perspectives of single-domain sensing data.** While multi-modal sensing data refers to data collected from heterogeneous sensor types, cross-domain sensing data refers to data obtained from a single sensor type but captured in different modalities or forms, such as RSS and Round Trip Time (RTT) from Wi-Fi APs, RGB images and depth maps from cameras, and range profiles and Doppler spectra from RADAR sensors. Because single-domain sensing data perceives the ambient environment from only a limited perspective, mobility analytics based solely on such data are often incomplete and less reliable. In contrast, cross-domain sensing enriches environmental perception by incorporating complementary representations of the same phenomenon, thereby improving the robustness and reliability of mobility analytics.

(3) **Issues of dataset staleness**. In passive sensing-based localization systems, the precise locations of passive sensors are often unknown. Therefore, extensive offline effort is typically required to associate sensing data with corresponding ground-truth locations in order to estimate users' accurate locations. However, data collection in passive sensing-based mobility analytics is not a one-time process. Instead, as existing IoT sensors leave or new sensors join, previously collected data may become stale. As a result, this staleness can degrade the reliability of mobility analytics potentially leading to inaccurate or misleading results.

1.2 Motivation

To deal with the aforementioned limitations and provide a comprehensive and reliable mobility analytics, this dissertation proposes a **collaborative** and **complementary** computing paradigm that integrates three key principles: 1) Complementation between multi-modal sensing data, where heterogeneous sensors are jointly utilized so that the strengths of one sensor compensate for the limitations of another. 2) Collaboration between cross-domain sensing data, where different forms of sensing data from the same sensor are incorporated to provide insights from different knowledge domains. 3) Multi-model-based sensing data generation, where Artificial Intelligence (AI) and non-AI methods are incorporated to generate sensing data in a timely manner, addressing the issues caused by dataset staleness.

Complementation between Multi-Modal Sensing Data

Spatial information can be passively extracted from various sensors that already exist in the environment, such as the signal strength from Wi-Fi APs, the light intensity from visible light lamps, or the depth maps captured by cameras. However, because these sensors are installed for purposes such as communication, illumination, or surveillance rather than mobility analytics, they may be removed, malfunction, or become unavailable at any time. Consequently, the loss of data from a single sensor type can degrade the performance of mobility analytics. In addition, each sensor perceives the physical environment only from a limited perspective, which often leads to incomplete or unreliable mobility analytics.

To address this challenge, this dissertation proposes complementation between multi-modal sensing data, which not only leverages the strengths of each sensing modality but also mitigates the impact of data loss when one type of sensor becomes unavailable. For example, Wi-Fi measurements provide spatial information but suffer from limited resolution due to multipath propagation. In contrast, IMU sensors capture fine-grained motion dynamics with high sensitivity but do not offer absolute location information. When these two types of sensing data are incorporated, Wi-Fi offers global spatial anchors while IMU data refines local movement, resulting in reliable mobility analytics that neither modality can provide independently. Furthermore, data loss may occur when cameras cannot detect users under occlusion. In such cases, wireless signals such as Wi-Fi or BLE, which typically penetrate or bypass common indoor obstacles and therefore are not significantly affected by visual occlusion, can provide complementary information to maintain mobility analytics performance. Conversely, when visibility is unobstructed, camera systems generally offer higher spatial accuracy than wireless sensing. Consequently, the joint utilization of wireless data and camera data enhances robustness by preventing failures that arise when a single sensing source becomes unavailable.

Collaboration between Cross-Domain Sensing Data

The same type of sensor can also perceive the environment from different perspectives depending on the form of data it produces. First, Wi-Fi-based localization systems exploit different signal features, including the RSS values [AAC18], the CSI [WWM18], and the RTT collected from ambient Wi-Fi APs [MRY+24; HYH20]. Moreover, visual data from cameras can also convey their perception of the world from different views. Compared with traditional localization based on RGB cameras [JLT+18], stereo cameras provide depth information that supports three-dimensional environmental perception [JKL18]. Furthermore, ambient light sources also offer multiple forms of information. While some studies use light intensity for location estimation [ZWZ+17; ZZ17], others leverage the modulation frequency of light to design localization systems [ZZ18].

In summary, a single type of sensor can generate different forms of data that convey information from different knowledge domains, each reflecting a distinct

viewpoint of the environment and providing its own strengths. Therefore, this dissertation proposes collaboration between cross-domain sensing data, leveraging the strengths of each data form to support a more comprehensive mobility analytics, especially in scenarios where only one type of sensor is available. For example, the RTT from Wi-Fi APs is easily accessible but provides only coarse spatial information due to multipath propagation. Similarly, light intensity measurements from ambient lamps can be readily obtained but do not provide orientation information. In contrast, the RTT measurements from Wi-Fi APs are more robust against multipath effects and can offer finer ranging information, but they require dedicated devices that support the Fine Time Measurement (FTM) protocol. Likewise, the modulation frequency of visible lights provides fine-grained spatial information but requires dedicated photodiodes for detection. Therefore, the collaborative use of RSS and RTT, as well as the collaborative use of light intensity and modulation frequency, enhances system robustness against data loss and produces more accurate spatial estimates for mobility analytics.

Multi-Model-Based Generation of Sensing Data

Datasets constructed for passive sensing-based mobility analytics are dynamic rather than static. This is because existing ambient sensors may be removed and new sensors may be deployed, causing the original dataset to become stale over time and degrading the performance of mobility analytics. Although continuously collecting new sensing data appears to be a straightforward method to maintain dataset freshness, it requires significant human effort and limits the scalability of mobility analytics services. To address dataset staleness, this dissertation proposes to regularly generate synthetic sensing data.

Approaches for generating synthetic data can generally be categorized into non-AI and AI methods. Non-AI methods are typically based on probabilistic modeling, where a limited amount of ground truth data is used to construct a mathematical representation of the data distribution, such as the estimation of means, variances, or covariances. In comparison, AI-based methods rely on neural network models that are trained using real data to learn high-dimensional features. Once trained, both the probabilistic model and the neural network model can be employed to generate extensive synthetic sensing data.

Each generation method has its own strengths and limitations. First, AI-based methods are effective at capturing nonlinear and complex features from sensing data. However, they generally require extensive training data, which incurs substantial human effort, and their performance degrades when the available data are insufficient. Probabilistic models, in contrast, often exhibit good extrapolation capability under limited training data because they rely on predefined kernel functions that implicitly model spatial correlations. Nevertheless, these kernel functions are usually fixed, and even subtle changes in the environment may significantly impair mobility analytics performance. Therefore, this dissertation proposes to jointly utilize different

generation models to generate passive sensing data, where the respective strengths and weaknesses of the models can be fully leveraged and compensated, thereby facilitating a more reliable and accurate mobility analytics.

1.3 Preliminary Work

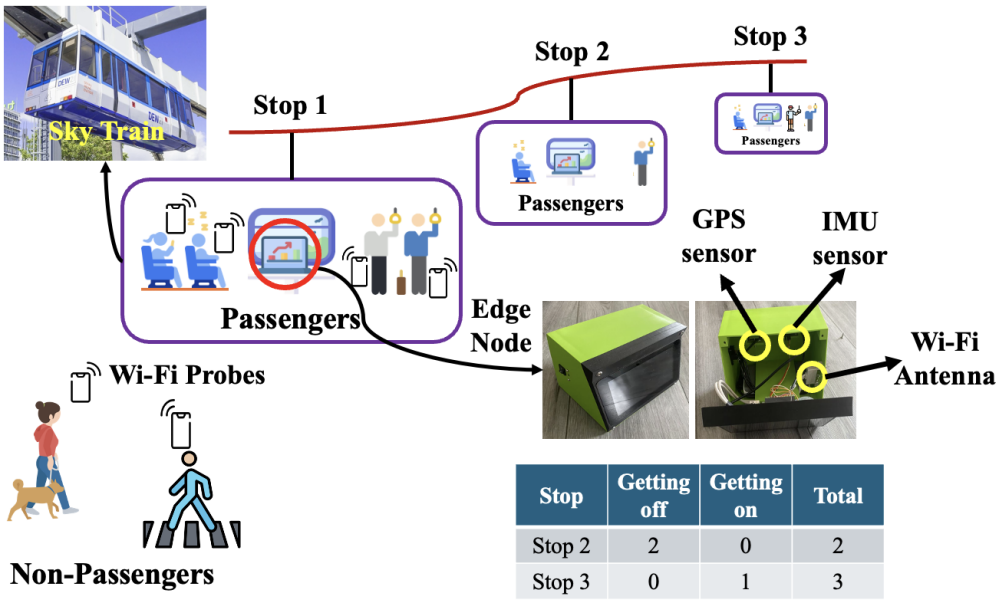
To demonstrate the feasibility of the proposed computing paradigm for mobility analytics, this dissertation first presents the collaborative and complementary use of multiple types of sensing data in the preliminary study. Next, this dissertation further validates the importance and effectiveness of sensing data collaboration and complementation by applying the proposed computing paradigm to mobility analytics in three different scenarios.

Collaboration and Complementation between GPS, IMU, and Wi-Fi

The public transportation system is one of the most essential components in modern society. Optimizing public transportation system, such as route planning, dynamic scheduling, and passenger flow estimation, plays an important role in enhancing passenger experience, reducing municipal expenditures, alleviating traffic congestion, and mitigating environmental pollution. In the preliminary work of this dissertation [WHD+20], multi-modal passive sensing data are jointly utilized to estimate passenger flow in a sky train system, as shown in Fig. 1.2. The key to passenger flow estimation is to analyze both vehicle mobility and passenger mobility, which supports real-time detection of the vehicle's movement status, stop events, and the number of passengers on board. To achieve this, this dissertation employs a Global Positioning System (GPS) sensor to obtain vehicle location information, an IMU sensor to capture vehicle motion patterns, and a Wi-Fi antenna to detect ambient Wi-Fi probe signals.

Vehicle mobility in this dissertation is analyzed using both IMU data and GPS data. The reliability of GPS measurements depends heavily on the Line-of-Sight (LOS) between the GPS sensor and the satellite. When the LOS is obstructed, for example when the vehicle enters a station or a tunnel, GPS data becomes unavailable for estimating the vehicle's position. To address this limitation, this dissertation designs a rule-based algorithm that fuses IMU and GPS measurements for vehicle motion status detection and stop detection. In this algorithm, the two sensing modalities complement each other, where GPS data provide absolute location information that IMU data lack, while IMU data compensate for periods of GPS unavailability.

Passenger mobility in this dissertation is investigated using Wi-Fi probe signals collected from the ambient environment. As a complement to vision-based approaches, Wi-Fi-based passenger mobility analytics does not compromise passenger privacy. In this work, a Wi-Fi antenna is employed to capture probe requests broadcast by nearby smart devices, which can be categorized into two groups: probes from



Stop	Getting off	Getting on	Total
Stop 2	2	0	2
Stop 3	0	1	3

Figure 1.2: Passenger flow estimation based on multiple sensing data.

passengers' devices and probes from non-passengers' devices, as shown in Fig. 1.2. To differentiate Wi-Fi probes from passengers and non-passengers, this dissertation designs a regression network that learns the relationship between features extracted from detected Wi-Fi probes and the actual number of passengers on the train. The well-trained model can estimate the number of passengers in real time using the Wi-Fi probes captured online.

In this preliminary work, the respective strengths of IMU data and GPS data are leveraged to complement their weaknesses, thereby facilitating more robust stop detection. In addition, the correlation between Wi-Fi probes and the actual number of passengers are correlated, supporting more reliable passenger tracking. The GPS sensor, the IMU sensor, and the Wi-Fi antenna are integrated into an edge node deployed on the train, where both the stop detection algorithm and the passenger tracking algorithm are executed in real time.

Collaboration and Complementation between BLE and Camera

Presence systems, which aim to monitor the presence of authorized individuals in specific environments, are important for ensuring public safety, improving productivity, and conserving energy. For example, during a pandemic, many airports, shopping malls, universities, and other public venues require visitors to present vaccination certificates before entering. By actively interacting with the environment, such as through manual QR code scanning, presence systems can support contact tracing. Presence systems are also widely used in large workplaces, where real-time monitoring of employee attendance enhances operational efficiency and

improves personnel safety. In addition, many facilities in the environment, such as lighting systems and elevator systems, benefit from presence-aware control, allowing access only to authorized individuals and thereby reducing energy consumption and improving safety.

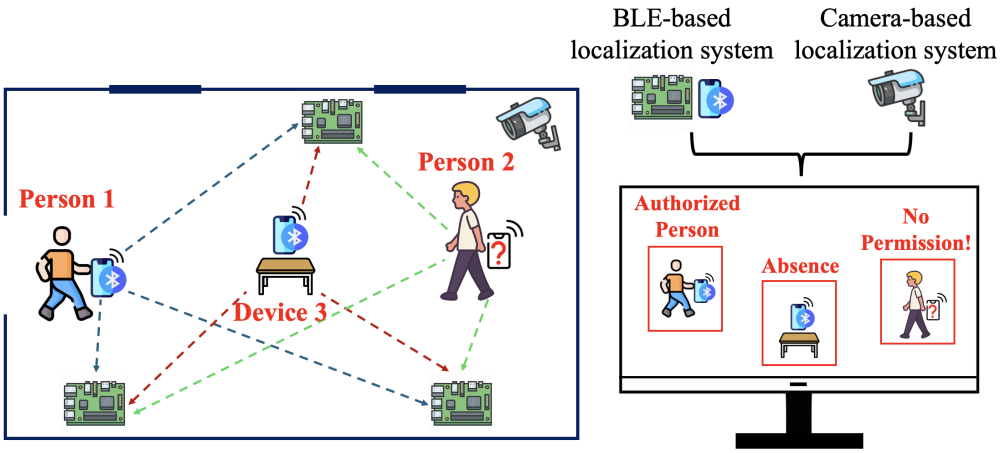


Figure 1.3: Presence detection based on multiple sensing data.

In our preliminary work [HWH+20], a presence system is developed that jointly utilizes wireless sensing data and visual sensing data to achieve automatic real-time monitoring of presence and absence. Visual sensing can "see" the types of objects in images, but cannot identify "whether they are allowed to enter the environment?" (e.g., whether they have an authorization certificate). Wireless sensing can "hear" the signals from objects (e.g., BLE tags or smartphones) that enable radio frequency, but cannot identify "what are there" (e.g., the types of objects). Three different scenarios are presented in Fig. 1.3 to exemplify the key idea behind our system. As shown in Fig. 1.3, Person 1 has an access certificate on his mobile device, Person 2 does not have an access certificate, and Device 3 has an access certificate but its owner is not in the environment. A reliable presence system is expected to be capable of detecting the authorized Person 1, the unauthorized Person 2, and the ownerless Device 3.

To achieve this goal, a cross-modal localization and detection system for indoor environments is developed. Wireless-based and vision-based localization are utilized collaboratively and complementarily to provide cross-validation of human presence. In the wireless-based localization system, a neural network model is trained to learn the correlation between the RSS of BLE signals and the locations where the signals are collected. In this work, devices continuously broadcast BLE beacons, and multiple receivers deployed in the environment capture these signals. The well-trained model can then estimate the real-time location of a device by analyzing the BLE signals received by the receivers. In the visual-based localization system, a depth camera continuously captures RGB images and depth maps. First, an object detection algorithm is executed to detect humans in the environment. Next,

both the RGB images and the depth maps are jointly used to localize all detected humans. Finally, a nearest neighbor algorithm-based method is designed to fuse the localization results of the two localization systems, leveraging their respective strengths to compensate for their limitations.

1.4 Goals, Challenges, and Contributions

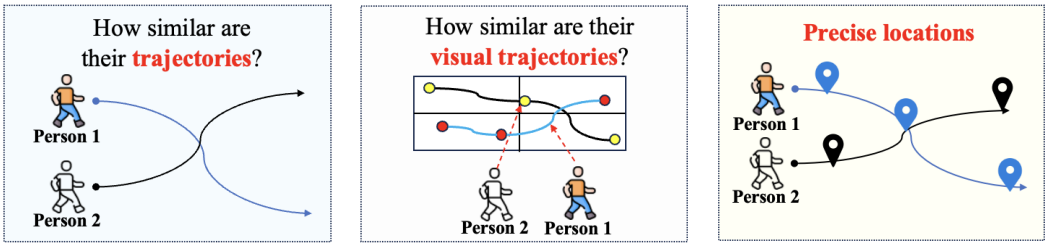
In the preliminary work, this dissertation incorporates multi-modal sensing data for mobility analytics in two different scenarios, demonstrating the feasibility of the proposed collaborative and complementary computing paradigm. To further verify the effectiveness of the proposed computing paradigm and address the three limitations explained in Section 1.1, three specific research objectives of mobility analytics are investigated in this dissertation, i.e., mobility trajectory comparison based on multi-modal sensing data, visual trajectory comparison based on cross-domain sensing data, and localization based on multi-model generation.

These three research topics are connected through the shared goal of characterizing human mobility at different spatial and analytical scales. As shown in Fig. 1.4(a), mobility trajectories describe how users traverse spaces or transition between areas and represent their movement at a macro level. Comparing users' mobility trajectories is to investigate the correlations between their macro-level behavioral patterns, supporting applications such as group detection, proximity detection, and anomaly or intrusion detection. Moreover, as shown in Fig. 1.4(b), visual trajectories represent changes in users' visual attention over time. Comparing users' visual trajectories is to analyze the correlations between their micro-level behavioral patterns within a confined spatial region, such as investigating whether they belong to the same attention group and whether their focus is on the same objects or regions. Furthermore, as shown in Fig. 1.4(c), analyzing users' precise locations provides the most granular level of mobility analytics, supporting location-dependent applications such as navigation, fitness-related services, and augmented reality gaming.

In summary, a top-down analytical hierarchy is established in this dissertation, ranging from macro-level human movement across spaces, to micro-level interaction behaviors within local regions, to fine-grained user locations. By addressing mobility across multiple spatial scales and sensing modalities, this dissertation aims to develop a comprehensive computing paradigm capable of capturing the full spectrum of human mobility.

Mobility Trajectory Comparison Based on Multi-Modal Sensing Data

Benefiting from the fast development of computing technology, the IoT-related services based on different types of sensors on mobile devices have penetrated our lives and are affecting us from different aspects. For example, the sensing data acquired from different sensors can be leveraged to facilitate more intelligent public transportation systems [LMS+18], improve emergency response and thus



(a) Mobility trajectory comparison. (b) Visual mobility trajectory comparison. (c) Users' precise locations.

Figure 1.4: Research goals of this dissertation.

enhance public safety [Wu18], infer users' social connections and behaviors [SBB+15], protect user privacy and improve mobile security [XYL+13], and augment other cyber-physical systems [WKT11].

Over the past few years, public health security has been severely compromised by the crises caused by pandemics. Social distancing during pandemics, if effectively monitored, is extremely important for preventing and avoiding further spread of diseases. For example, the level of social distancing can be determined by utilizing crowds' locations in a city [Una20], and the work in [Gov20] devises a method to detect human proximity to identify people in the same areas. However, privacy issues arise when location information is acquired [WL20]. While users may be willing to reveal their GPS location for location-based services, GPS-based localization may not be possible indoors due to the lack of LOS. In addition, proximity detection technology can only sense mobile devices within the communication range and unable to deeply comprehend users' trajectories. Furthermore, users' trajectories in the environment are usually random and unpredictable, and may even involve physical contact through social handshakes. It is challenging to quantify their level of social distancing on their trajectories in the physical world using only sensor data from the cyber world.

To tackle these challenges, this dissertation leverages the complementation between multi-modal sensing data from users' mobile devices to detect proximity between users. More specifically, this dissertation collects Wi-Fi data from ambient Wi-Fi APs based on the Wi-Fi antenna on users' devices, and acceleration data from the IMU sensor on users' devices. In this dissertation, we refer to the collected Wi-Fi data as **macro mobility** because it implies users' spatial locations and the acceleration data as **micro movement** because it portrays users subtle movements. In this dissertation, we refer to the collected Wi-Fi data as **macro mobility** because ambient Wi-Fi APs act as virtual landmarks and implies users' spatial locations. The acceleration data is referred to as **micro movement** because the IMU sensor captures body motions and portrays the subtle movements of a mobile user.

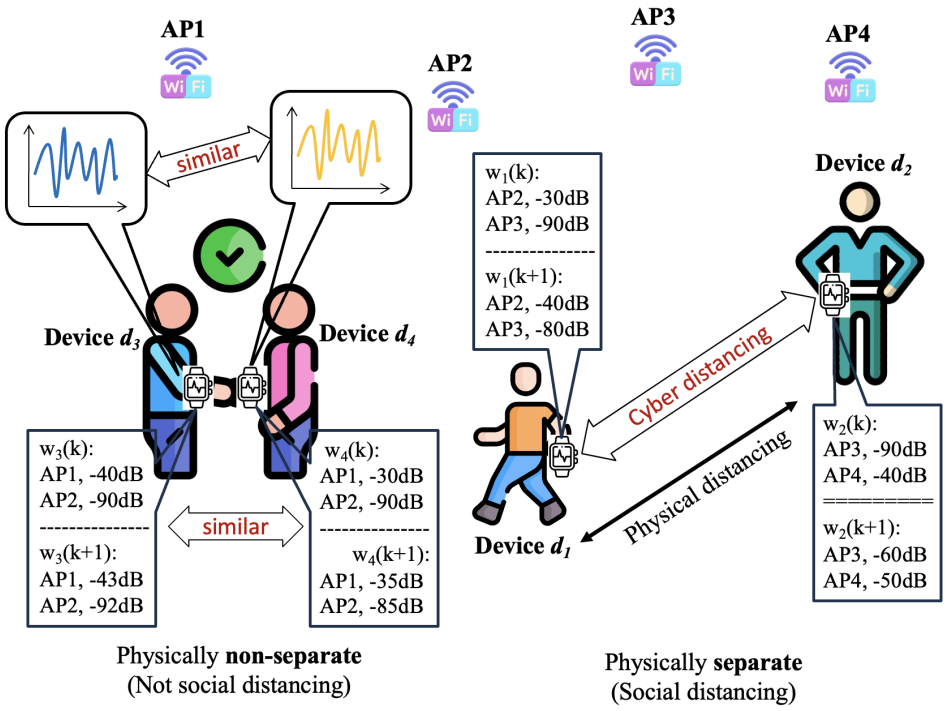


Figure 1.5: Proximity detection based on multi-modal sensing data.

This work aims to study the correlation between cyber distances behind multi-modal sensing data and physical distances created in real-world social interactions. To this end, we first quantify the **mobility similarity** between any two mobile users based on the virtual landmarks (i.e., Wi-Fi data) collected along their mobility traces. This dissertation considers two users as not social distancing (i.e., physically non-separate) when they are sharing similar mobility trajectories. In this situation, the Wi-Fi data collected by the two users are also similar, thus the cyber distance between their macro mobility data is short. On the contrary, two users are considered as social distancing (i.e., physically separate) when they are having completely different mobility trajectories. In this situation, the Wi-Fi data collected along their mobility trajectories are not similar to each other, and the cyber distance between their macro mobility data becomes large. Next, the **movement similarity** between the two users is quantified based on their collected acceleration data. This dissertation considers two users as not social distancing (physically separate) when they have the same micro-movements due to social physical contact (e.g., handshake behavior), as shown in Fig. 1.5. In this case, the accelerations of two users are similar, resulting in a shorter cyber distance between their movement data. Based on the quantification of mobility similarity and movement similarity, this dissertation proposes a cyber-physical social distancing system to check whether two users are physically non-separate or physically separate. For example, the designed system can determine that two users are physically separate when the computed mobility

similarity and movement similarity are smaller than a pre-set threshold, and vice versa. The proposed system is implemented as an application that can be installed on any device, as long as the device has a Wi-Fi antenna to collect data from Wi-Fi APs and an IMU sensor to collect data on its own acceleration.

The following contributions are made through the complementation between multi-modal sensing data (i.e., Wi-Fi data and accelerations) for proximity detection.

- **Complementary social computing**: Wi-Fi data implicitly indicates the spatial correlation between users and offers the opportunity to explore users' transitions across spaces. However, Wi-Fi data has a lower resolution and is unable to perceive small changes in mobility. IMU data, on the other hand, is very sensitive to small changes of users' motion, however, IMU data lacks the support of spatial information. Therefore, the complementation between multi-modal sensing data, i.e., Wi-Fi data and IMU data, can compensate for their respective weaknesses, enabling a more reliable proximity detection.
- **Location-less mobility analytics**: The proposed system is designed to detect proximity between users without the support of users' absolute locations. As the virtual landmarks (i.e., detected Wi-Fi APs) collected along users' mobility trajectories have been modeled into the proposed mobility signatures, the spatial correlation between users can be inferred by computing the mobility similarity between mobility signatures.
- **Correlating digital twins of mobile users in any environment**:The proposed metric links the physical distance between users and the cyber distance between their digital twins. Distance measurements based on RSS can be utilized to track groups and contacts in large public spaces, but may be challenging in small spaces because RSS is sensitive to many environmental influences, e.g., signal attenuation due to obstructions and multipath effects. Conversely, the proposed system is reliable in both large and small-scale environments, because our system considers both macro mobility and micro movements.
- **Low complexity for real-time applications**: Theoretical analysis and extensive experiments demonstrates that the complexity of computing both mobility similarity and movement similarity is low. As a result, the proposed system is well suited for real-time applications.

Visual Trajectory Comparison Based on Cross-Modal Sensing Data

Investigating the correlation between users' visual attention is of significance. First, online classes are becoming more popular because of the pandemic, monitoring the visual attention of students during online classes can remind distracted students and thus improve the quality of teaching and learning in online classes. Second, the user's visual attention can form different trajectories, which can be used as signals to control different home appliances in the smart home. Third, with the emergence of metaverse, exploring the correlation between users' visual attention can provide

different services for Virtual Reality (VR) in the future. Some recent research studies the visual attention of users by analyzing the images or videos captured by their wearables, aiming to understand users' behavior in different scenarios. For example, the works in [WS17; SMS+18] infer users' visual attention by analyzing the scenes they see based on scene cameras, while the works in [MC21; JHQ+16] investigate users' attention by examining the users' gaze using an eye camera.

However, limitations are observed in the existing methods. First, the scene camera-based approaches [WS17; SMS+18] are incapable of capturing the changes in visual attention caused by users' eyeballs. An example is given in Fig. 1.6, where person 1 is always concentrating on the computer during an online class while person 2 changes his attention from the computer to his smartphone by only moving his eyeballs. In this case, the scenes captured by the two persons are identical because their scene cameras always direct at the computer screen. In addition, the scene camera-based approaches involve privacy issues, as passers-by are inevitably captured as well. Furthermore, the gaze-based approaches [MC21; JHQ+16] are capable of perceiving the movement of users' eyeballs but provide no spatial information, e.g., users may share identical eye movements even in two completely different spaces.

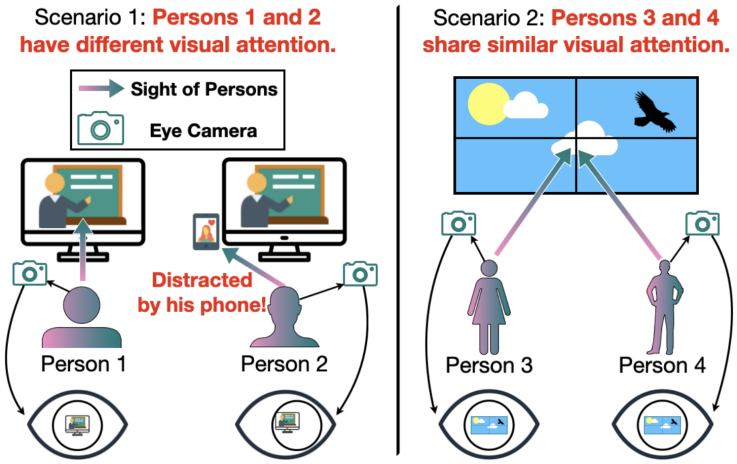


Figure 1.6: Visual attention detection based on cross-domain sensing data.

With the above limitations in mind, this dissertation investigates users' visual attention through the collaboration between cross-domain sensing data, i.e., the movements of human eyeballs and the light reflected in users' eyes. As is well known, human eyes act as a complicated optical system that perceives the light from the surrounding environment and reflects the light in the iris of the eye. When users' visual attention is in the same position, the light perceived by their eyes is identical and thus exhibits similar distribution in their irises. When users' visual attention is in different positions, the light from different environments generates different distributions in their irises. Therefore, the consecutive images (videos) containing

users' eyes represent the evolution of the light reflected in users' irises, which also implies changes in users' visual attention over time.

Therefore, this dissertation studies the correlation between users' visual attention by comparing the collected videos containing users' eyes. To achieve this, this work proposes a system for visual attention correlating based on light-tracing of human eyes. The proposed system models the collected video containing users' eyes as consecutive visual signatures, which represent the evolution of light from the perspective of the frequency domain. The high-frequency components are fast changes in light and the low-frequency components are slow changes in light. To explore whether two users share the same visual attention, the proposed system computes the similarity score of visual signatures between two users. The principle of quantifying the similarity score of any pair of visual signatures is to explore the overlap of their frequencies. The more similar frequencies two visual signatures share, the more similar the light distribution in users' eyes is. We refer to the average similarity between users' visual signatures as the visual attention similarity between two users. A higher visual attention similarity indicates that two users share the same visual attention and vice versa.

The following contributions are made through the collaboration between cross-domain sensing data from the same sensor, i.e., the movements of eyeballs and the light reflected in human eyes captured by an eye camera.

- **Collaborative visual attention correlating:** Most of the existing methods for visual attention correlating are based on scene cameras, which only capture the scenes seen by users. Some recent studies propose to capture eye movements to explore users' visual attention, because smart glasses are becoming more and more prevalent. This dissertation proposes a supplementary solution for visual attention correlating based on the light reflected in human eyes captured by an eye camera. The collaborative use of light reflected in users' eyes and the movement of users' eyes can reliably correlate users' visual attention without raising privacy concerns.
- **Correlation analysis between low-resolution videos:** In this dissertation, the video containing users' eyes is low-resolution where the most useful information is the light reflected in users' eyes. For each collected video, this dissertation creates consecutive visual signatures which are the representation of the video in the frequency domain. By investigating the overlap of similar frequencies between users' visual signatures, this dissertation provides an approach to analyze the correlation between the low-resolution videos.
- **Tolerant to data loss caused by human eyes:** Blinking and the rapid movements of human eyes may lead to data loss and consequently system failures. The system proposed in this dissertation investigates the current data along with upcoming data when capturing the user's eyes, thus mitigating the impairment of the system caused by blinking or rapid eye movements.

Localization Based on Multi-Model Generation

Indoor localization is important, particularly in environments where GPS data is unavailable due to the lack of LOS. Accurately predicting users' locations indoors enables seamless navigation in complex spaces, allows businesses in shopping malls to deliver precisely targeted advertisements, and enhances public safety by facilitating efficient emergency response. Many studies leverage the information provided by various sensing technologies for indoor localization, e.g., camera-based localization [DNX+18; YSR22; ZXW+19], visible light-based localization [LGY+21; HXH+18], IMU-based localization [SLJ+14; YMC+23], and BLE-based localization [CLC+20; YCC+21]. However, camera-based methods raise privacy concerns, visible light-based methods involve high deployment costs, and IMU-based methods suffer from error accumulation, necessitating collaboration with other sensing technologies. BLE- or UWB-based approaches [CFL+22; AUM+24], although more cost-effective than other approaches, incur significant hardware costs when deployed at scale.

Due to the prevalence of Wi-Fi technology, many recent studies exploit information captured from existing Wi-Fi APs to estimate the location of indoor users [YDV+13; SHZ+15; SWL+22; ZQZ+22; TWL+20; HYD+19]. On one hand, Wi-Fi-based localization raises no privacy concerns. On the other hand, Wi-Fi-based localization does not necessitate any additional infrastructure costs. Despite its advantages, Wi-Fi-based localization faces four main challenges: device heterogeneity, RSS fluctuation, high offline data collection costs, and dataset deterioration [LXY+24]. Device heterogeneity and RSS variation, which are already commonly discussed in many studies, can be addressed by designing robust fingerprinting techniques [YDV+13; SHZ+15]. To alleviate extensive efforts for Wi-Fi data collection, some studies propose to generate extensive synthetic Wi-Fi fingerprints by only using a smaller number of collected Wi-Fi samples [LYK+21; TR21; MS24]. Dataset deterioration is often attributed to changes in signal propagation due to furniture rearrangements or human movement within the environment [ZWZ+23]. Additionally, dataset deterioration also occurs when certain APs intermittently become undetectable [LXY+24]. To address this problem, semi-supervised learning frameworks are proposed in [ZWZ+23; LXY+24] to update the constructed radio maps from time to time.

However, the above studies overlook the underlying cause of dataset deterioration, namely the dynamic join or departure of Wi-Fi APs from the environment over time. As a consequence, the localization performance may be significantly impaired. To address this problem, an intuitive approach is to remove undetectable APs from the database and generate synthetic radio maps for newly observed APs. Recently, compared to traditional methods for radio map estimation [LH12; TZ21; LYK+21], Artificial Intelligence Generated Content (AIGC)-based approach has been receiving increasing attention because of its ability to extract reliable features from the high-dimensional Wi-Fi data. Still, some challenges remain to be addressed. For instance, Generative Adversarial Network (GAN)-based models are

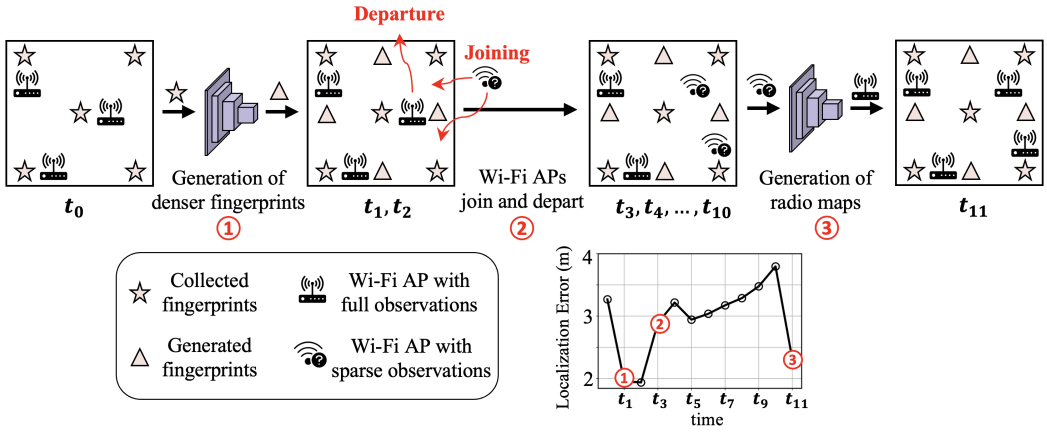


Figure 1.7: The radio map adaptation in this dissertation refers to 1) generating denser fingerprints for existing APs, and 2) generating radio maps for newly observed APs.

devised in [NCC+21; JP24; QTN+22] to generate extensive Wi-Fi fingerprints for ambient APs, based on the sparsely collected RSS observations from these APs. However, the synthetic fingerprints are rarely reliable when these models are applied to newly joined APs, because the extrapolation capability of GAN-based models is limited. The works in [ZWD23; ZAP+23] can generate full radio maps for wireless signal sources using generative models. However, transmitter locations are indispensable in [ZAP+23], and urban maps are required in [ZWD23], which is seldom available in reality.

To address these challenges, this dissertation proposes a collaborative and complementary framework for spatial reconstruction and localization based on radio map adaptation. Specifically, a radio map generator is developed by integrating a Gaussian Process Regression (GPR)-based model with a GAN-based model. On the one hand, the GAN-based model is capable of capturing complex and nonlinear features from RSS measurements, but it typically requires significant human effort to collect large-scale training data. When training data is scarce, the GAN-based model often exhibits limited extrapolation capability for radio map generation. On the other hand, the GPR-based model leverages spatial kernel functions to model correlations between locations and RSS values, achieving satisfactory extrapolation capability even with insufficient training data. However, due to its fixed kernel assumptions, its performance may degrade when applied to new environments with different spatial layouts and radio dynamics. Therefore, this dissertation combines GPR and GAN to leverage their complementary strengths for radio map generation. The proposed GPR-GAN generator is leveraged for radio map adaptation from two perspectives, as demonstrated in Fig. 1.7. First, during the data collection phase (time t_0 in Fig. 1.7), sparsely collected Wi-Fi samples are used to train the GPR-GAN generator, which is subsequently employed to synthesize extensive Wi-Fi

measurements to densify the radio maps. Second, during the localization phase (time t_1 to t_{10} in Fig. 1.7), the GPR-GAN generator is utilized to generate radio maps for newly observed Wi-Fi APs, thereby alleviating dataset staleness caused by radio environment dynamics. Extensive results show that the proposed framework is temporally robust, with its localization error increasing only slightly from 1.750 m to 2.170 m over one year, while two State-Of-The-Art (SOTA) methods degrade significantly, from 1.955 m and 2.172 m to 3.402 m and 2.770 m, respectively. The following contributions are made through the collaborative and complementary generation of Wi-Fi fingerprints.

- **Collaborative and complementary generation with dual-model:** The proposed framework for radio map generation is developed through the joint collaboration of a GPR model and a GAN model. By embedding the GPR model into the adversarial learning process, this dissertation collaboratively and complementarily integrates the interpolation ability of GPR and GAN's ability to learn useful features from high-dimensional data.
- **Temporal-robust localization against dataset deterioration:** This dissertation develops a radio map generator, which exploits sparse RSS observations of newly detected APs to generate full radio maps for these APs. Consequently, the time-varying deterioration of the Wi-Fi dataset is alleviated through radio map adaptation, thus yielding a time-robust localization system.
- **Spatial-resilient fingerprinting technique:** This dissertation additionally devises a novel fingerprinting algorithm, where the fingerprints are created by exploring the fine-grained RSS correlation between APs. While some APs fail to provide reliable data because of environmental changes, the devised fingerprinting technique exploits information from other APs for compensation.
- **Labor-saving creation of denser radio maps:** When Wi-Fi samples are collected only at sparse RPs, the well-trained radio map generator can also be applied to generate extensive synthetic Wi-Fi data, thereby greatly reducing the expensive labor costs required for dataset collection.

The rest of this dissertation is structured and organized as follows. In Chapter 2, this dissertation introduces some existing state-of-the-art research on proximity detection based on multi-modal sensing data, visual attention detection based on cross-domain sensing data, and indoor localization based on multi-model generation of Wi-Fi data. In Chapter 3, this dissertation elaborates on how to devise a proximity detection system based on the complementation of multi-modal sensing data, and demonstrates the performance of the devised system through extensive experiments. In Chapter 4, this dissertation designs algorithms to investigate the correlation between users' visual attention based on the collaboration of cross-modal sensing data. Similarly, extensive experiments are conducted in this chapter to validate the performance of the designed algorithms. In Chapter 5, this dissertation proposes a Wi-Fi-based indoor localization framework, where multiple generation models are jointly utilized to generate denser sensing data. Next, based on the original data and

the synthetic data, this dissertation designs a robust Wi-Fi-based indoor localization system, and verifies through extensive experiments that the system outperforms the state-of-the-art. Finally, this dissertation summarizes the importance of multi-modal sensing data, cross-domain sensing data, and multi-model-based generation of sensing data in mobility analytics in Chapter 6. Furthermore, this dissertation also prospects potential future work in this chapter.

1.5 Author's Contribution to this Dissertation

According to §10(2) of the "Promotionsordnung der Fakultät für Informatik der Technischen Universität Dortmund vom 29. August 2011", the dissertation must include statements about the author's contributions that resulted from cooperations with others. Below, the contributions and statements are listed.

- The content of the preliminary work of this dissertation is divided into two parts, which have been published in [WHD+20] and [HWH+20], respectively. All authors equally contributed to the work in [WHD+20], where the idea was proposed by Fang-Jing Wu, the algorithm design and experiments were jointly completed by me together with other authors, the implementation was completed by me, and the writing of this work was completed by Fang-Jing Wu. The idea of [HWH+20] was proposed by Fang-Jing Wu, the algorithm design and experiments were jointly completed by me and other authors, and the writing of this work was mainly completed by me.
- The content in Chapter 3 has been published in [HW21], which was completed collaboratively by Fang-Jing Wu and me. The idea of this work was proposed by Fang-Jing Wu and written together by Fang-Jing Wu and me. The algorithm design, experiments, and implementation of this work were completed by me, and Fang-Jing Wu offered many valuable suggestions and guidance throughout the process as my supervisor.
- The content in Chapter 4 was completed collaboratively by Fang-Jing Wu and me, which is currently in the submission stage [HW25b]. The idea of this work was proposed by Fang-Jing Wu and me after many discussions. The work was written collaboratively by Fang-Jing Wu and me. The algorithm design, experiments, and implementation of this work were completed by me. Fang-Jing Wu provided me with many valuable suggestions and guidance throughout the process as my supervisor.
- The content in Chapter 5 was completed collaboratively by Fang-Jing Wu and me, which is currently in the submission stage [HW25a]. The idea of this work was proposed by Fang-Jing Wu and me after many discussions. The work was written collaboratively by Fang-Jing Wu and me. The algorithm design, experiments, and implementation of this work were completed by me. Fang-Jing Wu provided me with many valuable suggestions and guidance throughout the process as my supervisor.

Chapter 2

Related Work

This dissertation proposes a collaborative and complementary computing paradigm for passive sensing-based mobility analytics. In the proposed computing paradigm, this dissertation detects proximity based on the complementation between multi-modal sensing data, investigates users' visual attention based on the collaboration between cross-domain sensing data, and designs a indoor localization system through the multi-model-based generation of Wi-Fi data. In this chapter, this dissertation introduces related work on mobility analytics in the above three scenarios.

Passive Sensing-Based Proximity Detection

The essence of proximity detection based on passive sensing data is to characterize human physical mobility based on different sensing data, and then infer humans' proximity by comparing the similarity between their mobility characteristics. Investigating humans' physical mobility based on sensing data is challenging. This dissertation first categorizes existing research on humans' physical mobility analytics into the following types: 1) vision-based methods, 2) localization-based methods, and 3) wireless signal-based methods. Then, this dissertation analyzes the weaknesses of these methods compared to our method.

Vision-based approaches utilize features of human mobility extracted from video frames for mobility analytics. The works in [GCR12; SCC16] detect groups of crowds by clustering mobility trajectories on continuous video frames. Social groups on images collected in a commercial space are identified in [MPG+10], where the identified social groups such as families, couples, and friends, follow a Poisson distribution. The work in [CWB+16] generates extensive synthetic crowd videos with the help of labeled ground truth in pedestrian counts and flows, significantly reducing data labeling efforts. Contrary to vision-based methods, our method does not require additional cameras and therefore does not violate user privacy. Furthermore, our method does not require extensive labeled data that is usually necessary for vision-based methods.

Localization-based approaches utilize relative or absolute locations to investigate human mobility. The works in [YHS+09; Fan86; Al-12; HLC+05] employ GPS locations collected by mobile devices to track human mobility. However, GPS data is usually not available in indoor environments because of the absence of LOS. Therefore, the earlier works in [YLL09; XCZ08] deploy additional and dedicated infrastructure with wireless beacons for indoor localization. The work in [ZXZ+14] jointly utilizes spatio-temporal Wi-Fi information to create indoor fingerprints for

localization. BLE beacons are set up in [MMS+17] to estimate users' locations instead of using Wi-Fi signals. The work in [VGQ16] leverages multi-modal wireless signals from pre-installed BLE beacons and Wi-Fi APs to localize mobile users based on trilateration. The work in [ZZX+15] divides the targeted field into multiple sub-regions, where each sub-region is encoded by a sequence of anchor nodes based on RSSIs for reducing the additional efforts at creating radio maps. The works in [YDV+13; WYL+12; AMH18] allow mobile users to participate in the creation of radio maps, aiming to reduce extensive efforts caused by prior site survey. In localization-based methods, location information is usually acquired at a high cost by deploying dedicated sensor networks on a large scale. Alternatively, location information can also be obtained offline through prior site surveys. Our method does not require any prior site surveys for offline data collection, and enables a plug-and-play approach.

Wireless signal-based methods leverage the information from ambient Wi-Fi APs or BLE beacons for mobility analytics. The work in [SW17] detects human mobility for the detection of static groups by setting up BLE sniffers in an indoor environment. The work in [ÇDG+18] monitors the levels of occupancy indoors, where Wi-Fi sniffers are utilized to collect Wi-Fi probe packets. Based on the Wi-Fi data extracted from multiple Wi-Fi sniffers' probe packets, this work leverages trilateration to estimate the user's location. A regression-based model is employed in [LRC19] to predict the level of occupancy indoors based on both BLE advertising packets and Wi-Fi probe request packets. The work in [WS18] incorporates Wi-Fi sniffers and stereoscopic cameras to develop a regression model for the estimation of crowd sizes. The work in [DYY+17] devises algorithms to identify the relationship among group members such as the left-right relationship or the leader-follower relationship, that can be inferred from the face-to-face/back-to-back interaction in a group. The work in [SCL+18] identifies shopping groups in shopping malls by analyzing the Wi-Fi associated data. The work in [HS19] detects whether two persons are moving along a similar trajectory by modeling crowd flows in graphs. In this work, the popularity of Wi-Fi APs is considered for the computation of mobility similarity between users. A higher popularity of a Wi-Fi AP means that this Wi-Fi AP can be detected by many people, indicating that this AP carries less important information. Compared to these research efforts, our work analyzes device-to-device social closeness along dynamic mobility traces and human interactions resulting from physical contact without relying on pre-deployed network sniffers.

Passive Sensing-Based Visual Attention Detection

We investigate the correlation between users' visual attention by comparing the captured videos containing their eyes. In this section, we categorize the research for video comparison into three different categories, which are 1) traditional feature-based approaches, 2) traditional hashing-based approaches, and 3) deep learning-based approaches.

In **traditional feature-based approaches**, features are extracted from the image frames of a video as video representations. The correlation between videos is quantified by computing the similarity between their extracted representations. For example, the work in [LLX12] first extracts the Scale-Invariant Features (SIFT) [Low99] from the image frames of videos. A Singular Value Decomposition (SVD) based algorithm is then designed in [Low99] to quantify the similarity between videos based on the extracted features. The Speeded-Up Robust Features (SURF) [BTG06], which are the fast version of SIFT, are extracted in [RR13] as the video representations. A Dynamic Time Warping (DTW) based algorithm is then designed in [RR13] to match the extracted SURF features between videos. The work in [RCW+12] creates visual features for videos using the Histogram of Optical Flow (HOOF) features. Then, the similarity between videos is computed by aligning these visual signatures. The work in [CCL15] extracts features by combining SURF and HOOF. The features extracted in [WHN07] are the color histograms, and the work in [SYW+10] extracts the Local Binary Pattern (LBP) features which are based on the grey-scale intensity of image frames. The work in [YC09] extracts the Markov stationary features [LP17] which are the extension of color-histograms. The work in [LBG+06] pays more attention to corner information, and thus extracts features using Harris detector [HS+88].

Traditional hashing-based approaches generate hash codes for videos and measure the similarity between videos by analyzing the correlation between their hash codes. For example, the work in [GIM+99] leverages Locality-Sensitive Hashing (LSH) to generate hash codes for videos. The work in [KSH+04] first extracts features from videos using a Principal Component Analysis (PCA) based SIFT method, the hash codes are then generated for these features using LSH. The work in [YCJ12] extracts SURF features from videos and then generates hash codes for the extracted features. The work in [LM12] regards videos as order-3 tensors, i.e., 2-D content of image frames and temporal evolution as the third dimension. Then, the hash codes are generated through multi-linear subspace projections of the video tensor, which is obtained with PARAFAC factorization. Discrete Cosine Transform (DCT) based hashing algorithms are also widely used for video representation. The work in [EFW10] applies 3D-DCT to the videos, the lower-frequencies components are then extracted as the hash codes of videos. 3D-DCT and Random Base Transform (RBT) are combined to generate hash codes for videos in [CSM06], and the hamming distance between hash codes is computed to investigate the similarity between videos. The work in [KB17] designs a hash algorithm by combing DCT and Discrete Cosine Transform (DST). In work [DWB19], a Temporal Match Kernel (TMK) and DCT-based perceptual hashing algorithm are jointly leveraged to quantify the similarity between videos.

Deep learning-based approaches extract features from intermediate convolutional layers of Convolutional Neural Network (CNN) and aggregate the extracted features as video representations. In work [KPP+19], the features extracted from CNN convolutional layers are aggregated based on pooling layers

and PCA technology. Tensor dot similarity and Chamfer similarity are then jointly exploited to quantify the similarity between videos. The work in [KPP+17] aggregates features based on a bag-of-word method and then computes the cosine distance between video representations. Instead of extracting features frame by frame, the work in [LLT+16] extracts the features directly from the entire video. The extracted features are then fused through fully convolutional layers to learn the temporal and discriminative information. The temporal information is also learned from different Recurrent Neural Networks (RNNs). For example, the work in [SZL+18] proposes a novel Binary Long Short-Term Memory (BLSTM) to obtain temporal information in a video. In addition, the works in both [WNS+19] and [LCL+19] also take the advantage of Long Short-Term Memory (LSTM) to investigate temporal information after extracting features from the intermediate convolutional layers.

The above research is not applicable to our work for three reasons. Firstly, the videos extracted from human eyes contain limited color information, which only consists of a large area of dark background and a small area of bright foreground. The dark background is from human irises, while the bright foreground originates from external light sources. If we generate hash codes for such images or videos, the useful information in the generated hash codes is sparse due to the large area of dark background. Secondly, the distinctive features in the videos extracted from human eyes are only the edges between the dark background and the bright foreground. The correlation between videos is hardly explored based on only edge features. Thirdly, objects are difficult to be detected in the videos because the videos captured from the eyes are of low resolution.

Passive Sensing-Based Indoor Localization

In this work, we categorize Wi-Fi-based indoor localization into (1) fingerprint-free Wi-Fi Localization, (2) traditional fingerprint-based Wi-Fi localization, and (3) fingerprint augmentation-based Wi-Fi localization.

Fingerprint-free Wi-Fi localization. Fingerprint-free Wi-Fi localization is usually achieved based on AoA-based methods or ToA-based methods. In AoA-based approaches, the antenna layout generally plays a vital role in localization performance. The works in [TLT+21; TWL+21] first design self-calibration methods to effortlessly optimize the antenna layout, where non-linear antenna layouts always exhibit superiority over linear antenna layouts. Next, different AoA-based methods are proposed for localization based on CSI information. The work in [ZZL+23] designs a confidence-based localization algorithm to choose the highest-scoring AoA estimate as the final localization result, where only the most reliable CSI measurements are selected for fingerprint construction. Due to the recent prevalence of FTM in commercial Wi-Fi chips, ToA information can be estimated more accurately based on RTT measurements [YCS+22; Cho22; ZWC+23; MWP+20]. The work in [MWP+20] first measures the distances between the target user and APs based on

FTM technology, and then introduces a clustering-based trilateration supported by weighted concentric circle generation (WCCG) for indoor localization. In addition, different types of Kalman filter algorithms are designed in [YCS+22; ZWC+23], which jointly utilize the distances estimated by FTM and the IMU sensor data from mobile devices for localization. The work in [Cho22] enhances the FTM-ranging results with the aid of IMU sensors.

Traditional fingerprint-based Wi-Fi localization. In this category of methods, the online perceived Wi-Fi data is generally matched with the offline constructed radio maps for localization. To address the issue of RSS uncertainty caused by indoor signal fading, the matching process in [ZLT+21] incorporates various hypothesis tests for localization, e.g., the Jarque-Bera test, Mann-Whitney U-test, and T-test. The work in [SWL+22] constructs fingerprints by synergizing RSS and FTM technology, which complements the limitations of single-modal data and improves localization accuracy. The works in [ZQZ+22; TWL+20] utilize the Wi-Fi CSI to construct fingerprints. In the online stage, a CNN-based model is designed in [ZQZ+22] to explore the correlation between locations and constructed fingerprints, while a LSTM-based model is devised in [TWL+20] for indoor localization. The fingerprints in [LYH+22] are established by synergizing a neural network and a genetic algorithm. Next, a nonlinear optimization algorithm is applied to the constructed fingerprints for indoor localization.

Fingerprint augmentation-based Wi-Fi localization. In this category of methods, a small number of Wi-Fi samples are usually collected offline, and then extensive synthetic Wi-Fi fingerprints are generated by correlating the collected Wi-Fi data and the target environment. The works in [MS24; LYK+21] generate denser fingerprints by incorporating sparse Wi-Fi samples collected offline and urban maps containing environmental geometric information. The GAN-based frameworks are proposed in [NCC+21; QTN+22] to generate a large number of fingerprints, from which the most reliable fingerprints are then selected by dedicated selection algorithms. Similarly, denser fingerprints are generated in [JP24] and in [ZCL+20] using a devised LSTM-GAN model and a GPR-GAN model, respectively, both of which are also trained using only sparse Wi-Fi samples collected offline. The works in [ZWD23; ZAP+23] jointly utilize offline collected Wi-Fi samples and signal transmitters' locations to train generative models, which are then employed to estimate the radio map of each transmitter in the target environment. The work in [ZSZ+24] leverages interval data analysis to generate APs' radio maps based on their sparse fingerprints collected offline. The work in [ZWZ+23] designs a few-shot learning framework to generate fingerprints, while a semi-supervised learning framework is proposed in [LXY+24] for radio map updating. The above research generates RSS-based Wi-Fi fingerprints, while the works in[LQL+19; WYW+21; CC20] employ generative models to generate CSI-based Wi-Fi fingerprints.

Offline collecting Wi-Fi samples to construct the fingerprint dataset is a two-sided process, offering both advantages and limitations. On one hand, fingerprint-free localization does not necessitate offline data collection, thus avoiding the associated

labor costs and enabling plug-and-play. On the other hand, fingerprint-based localization learns the correlation between the target environment and Wi-Fi information through the created fingerprints, thereby improving localization performance. To alleviate expensive labor costs while ensuring localization performance, fingerprint augmentation-based localization is proposed to generate extensive synthetic fingerprints using a small number of collected Wi-Fi samples. However, the dynamic join and departure of Wi-Fi APs are not taken into account in the above studies, leading to a potential degradation of localization performance over time. In addition, urban maps and AP locations are usually hard to acquire in reality. The framework proposed in this work aims to solve these problems.

Chapter 3

Mobility Trajectory Comparison Based on Multi-Modal Sensing Data

3.1 System Model

In this work, the social closeness among users in the physical world is investigated by studying the correlation between their cyber-world mobility trajectories. This work creates the cyber-world mobility trajectory for each user using the sensing data collected by their smart devices. To this end, we first assume that a dedicated mobile sensing application is installed on the device of each user in advance. On the one hand, this application perceives the information from ambient Wi-Fi APs every T_w seconds. On the other hand, it collects accelerations from the device's IMU sensor every T_a seconds. This work divides time into intervals with a fixed length of I, i.e., the mobility analysis is always executed based on the sensing data collected within I seconds. For a certain time interval, we denote by $\mathbb{W}_i = \{w_i(k)|k > 0\}$ and $\mathbb{A}_i = \{(x_i(k), y_i(k), z_i(k))|k > 0\}$ the Wi-Fi data and accelerations collected by user d_i, respectively. Note that, $w_i(k)$ contains the MAC addresses of detected Wi-Fi APs and their corresponding RSS, and $x_i(k)$, $y_i(k)$, $z_i(k)$ represent the accelerations of d_i along the x-, y-, and z-axis at the k-th measurements. As a result, this work creates d_i's mobility trajectory based the sensing data $(\mathbb{W}_i, \mathbb{A}_i)$ collected by d_i during any time interval I, where $|\mathbb{W}_i| \neq |\mathbb{A}_i|$ because the frequency of sampling Wi-Fi data is significantly higher than sampling accelerations.

Assuming that the sensing data $(\mathbb{W}_i, \mathbb{A}_i)$ and $(\mathbb{W}_j, \mathbb{A}_j)$ are collected by two users d_i and d_j, this work computes the cyber-world distance between d_i and d_j by quantifying the similarity between their sensing data. Because this work models the mobility trajectory for each user based on their sensing data, a higher similarity between $(\mathbb{W}_i, \mathbb{A}_i)$ and $(\mathbb{W}_j, \mathbb{A}_j)$ means that d_i and d_j are sharing similar mobility trajectories. As a result, the cyber distance between d_i and d_j is also shorter. Similarly, a larger cyber distance indicates a lower similarity between $(\mathbb{W}_i, \mathbb{A}_i)$ and $(\mathbb{W}_j, \mathbb{A}_j)$. To achieve this, this works first separately compute the similarity between $\mathbb{W}_i$ and $\mathbb{W}_j$, and the similarity between $\mathbb{A}_i$ and $\mathbb{A}_j$. The $\mathbb{W}_i$-$\mathbb{W}_j$ similarity implies whether d_i and d_j are in close proximity, while the $\mathbb{A}_i$-$\mathbb{A}_j$ similarity indicates whether d_i and d_j are taking consistent movements, such as handshakes. Next, the $\mathbb{W}_i$-$\mathbb{W}_j$

similarity and $\mathbb{A}_i$-$\mathbb{A}_j$ similarity are fused to estimate the cyber distance between d_i and d_j.

This work leverages the proposed cyber distance to check whether users are "physically separate" or "physically non-separate", that is a significant use case for mobility analysis. In this work, a shorter cyber distance between d_i and d_j implies that d_i and d_j are "physically non-separate" (i.e., not social distancing), and a larger cyber distance indicates that d_i and d_j are "physically separate" (i.e., social distancing), as shown in Fig. 1.5.

3.1.1 System Overview

The system designed in this work is composed of the following four phases: 1) collection of sensing data, 2) creation of multi-modal signatures, 3) computation of cyber-world distances, and 4) detection algorithm, as shown in Fig. 3.1.

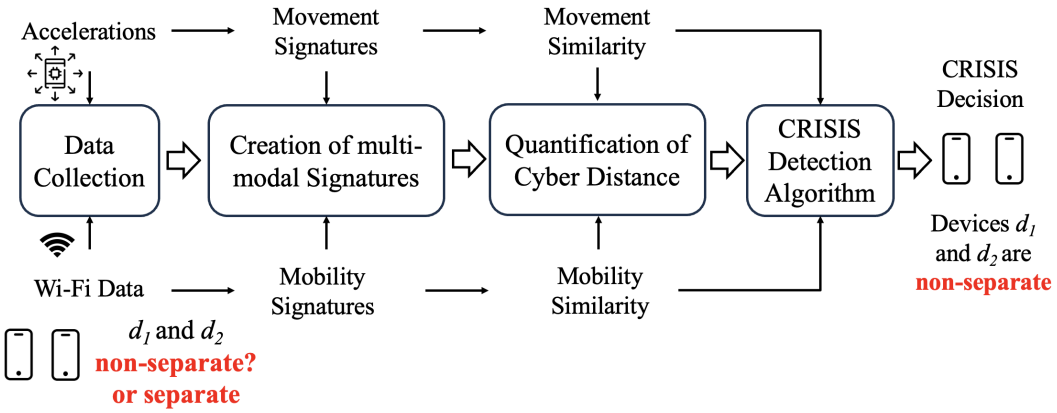


Figure 3.1: An overview of the proposed system.

In the first phase, the mobile sensing application captures ambient Wi-Fi APs and records accelerations from the smartphone's built-in IMU sensor. The collected Wi-Fi data includes the unique identifier of each detected AP and its corresponding RSS measurement. To address privacy concerns, this work employs a data anonymization mechanism, namely the SHA3-256 hashing algorithm [LFZ+16], to anonymize the IDs of Wi-Fi APs and users' devices. In addition, data collection is strictly permission-based, and sensing begins only after users install the application and provide explicit consent. Therefore, each user is associated with a unique ID, and all collected Wi-Fi and sensor samples are tagged with both this user ID and a timestamp. All data collected in this phase are forwarded to the second phase, where multi-modal signatures are created. Here, the Wi-Fi measurements and IMU accelerations collected by each user over time are transformed into mobility signatures and movement signatures. Since these multi-modal signatures are sequentially continuous, they are regarded as the user's mobility trajectory during that period. In the third phase, the system computes the cyber-world distance between different

users by quantifying the similarity of their mobility trajectories. In the final phase, the detection algorithm determines whether users are "physically separate" or "physically non-separate" based on the computed cyber distance.

3.2 Creation of Multi-modal Signatures

3.2.1 Creation of Mobility Signatures

This section introduces the technical details of creating mobility signatures for the collected Wi-Fi data. The key principle behind the creation of mobility signatures is to assign greater weights to important APs and lower weights to less important APs. In this work, the AP's importance is given based on their RSS values. When a user detects an AP with a strong RSS, it means the AP is in closer proximity to the user, the probability of LOS is higher, and the impact of signal fading is lower. Therefore, giving more attention to these APs with higher RSS values can effectively reduce the uncertainty in the collected Wi-Fi data.

Assuming that $w_i(k)$ is the Wi-Fi data collected at the k-th scan, a mobility signature is created for $w_i(k)$ by assigning non-negative weights to the APs detected in $w_i(k)$. We denote by $\widetilde{w}_i(k) = \{(c_1, r_1, \sigma_1), (c_2, r_2, \sigma_2), \ldots, (c_l, r_l, \sigma_l), \ldots\}$ the created mobility signature, where any AP (AP-l) in $\widetilde{w}_i(k)$ can be represented by a 3-tuples list (c_l, r_l, σ_l), i.e., a list containing the MAC address c_l, the RSS value r_l, and the weight σ_l to be assigned for AP-l. Note that, all APs in $\widetilde{w}_i(k)$ are sorted by their RSS values, i.e., $r_1 \leq r_2 \leq, ..., \leq r_l$. The weight σ_l assigned to AP-l is computed according to Eq. (3.1) and Eq. (3.2), the key idea of that is to progressively reduce the importance level of each AP.

$$R(l+1) = R(l) - \frac{r_l - r_{l+1}}{\delta_w},\ l = 1, \ldots, |w_i(k)| - 1, \tag{3.1}$$

$$R(1) + R(2) +, \ldots, +R(|w_i(k)|) = 1. \tag{3.2}$$

As an example, the importance level of AP-1, denoted by $R(1)$, is the strongest in $\widetilde{w}_i(k)$ because AP-1 has the strongest RSS after sorting. Next, the importance level of AP-2, denoted by $R(2)$, can be represented by subtracting $\frac{r_1 - r_2}{\delta_w}$ from $R(1)$ according to Eq. (3.1), where $r_1 - r_2$ describes the RSS correlation between AP-1 and AP-2 and δ_w is the empirically determined decreasing rate. In this way, the importance level of any AP can be represented by $R(1)$ if the RSS correlation between AP-1 and this AP is found by recursively executing Eq. (3.1). Subsequently, $R(1)$ is obtained by substituting the importance level of each AP into Eq. (3.2), and $R(2), R(3), ..., R(l), ...$ can be also obtained by substituting the computed $R(1)$ into Eq. (3.1). Finally, the weight assigned to each AP is given by converting the importance level of each AP to non-negative value, as shown by $\sigma_l = \max\{R(l), 0\}$, $l = 1, \ldots, |w_i(k)|$.

An illustrative example of the creation of mobility signatures is presented in Fig. 3.2(a). As can be seen, eight Wi-Fi APs are detected at the first Wi-Fi measurements $w_i(1)$. To transform $w_i(1)$ to its mobility signature $\widetilde{w}_i(1)$, weights need to be assigned to the eight APs based on their RSS. To this end, the importance level of each AP is first represented with $R(1)$ based on Eq. (3.1), e.g., $R(2) = R(1) - \frac{(-60dB) - (-66dB)}{\delta_w}$ and $R(3) = R(1) - \frac{(-60dB) - (-66dB)}{\delta_w} - \frac{(-66dB) - (-69dB)}{\delta_w}$, and so on. Next, $R(1) = 0.30$ can be computed when $R(2), R(3), ..., R(l), ...$ are substituted into Eq. (3.2). Finally, the weights for the eight Wi-Fi APs are computed as $\sigma_1 = 0.30$, $\sigma_2 = 0.23$, $\sigma_3 = 0.19$, $\sigma_4 = 0.09$, $\sigma_5 = 0.09$, $\sigma_6 = 0.06$, $\sigma_7 = 0.03$, $\sigma_8 = 0.01$, based on the computed $R(1)$ and Eq. (3.1).

3.2.2 Creation of Movement Signatures

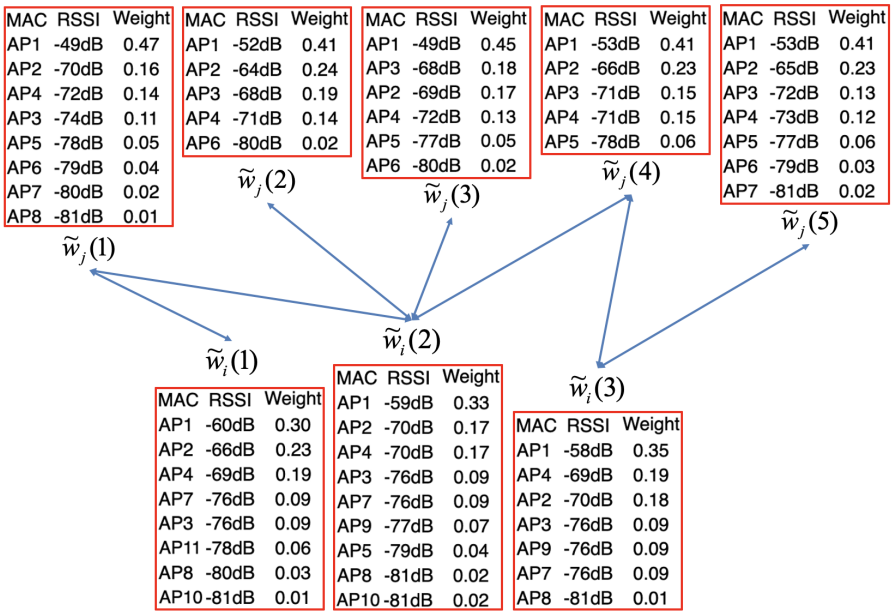
The key to generating movement signatures is to model human motion in both time and frequency domains, which can be implemented by initially processing the collected acceleration data. Given $\mathbb{A}_i = \{(x_i(k), y_i(k), z_i(k)) | k > 0\}$ collected by d_i during a certain time interval, the accelerations are converted into $\widetilde{\mathbb{A}}_i = (\widetilde{\theta}_i, \widetilde{\phi}_i)$, where $\widetilde{\phi}_i = \{\widetilde{b}_i(p) | p \in [0, F_{max}]\}$ is the movement signature in the frequency domain and $\widetilde{\theta}_i = \{\widetilde{a}_i(k) | k > 0\}$ represents the movement signature in the time domain. In this case, $\widetilde{a}_i(k)$ is the filtered acceleration of k-th raw accelerometer measurement, and $\widetilde{b}_i(p)$ is the magnitude in frequency $p \in [0, F_{max}]$ after applying the Fast Fourier Transform (FFT) to $\widetilde{\theta}_i$. In this work, F_{max} is set to 10 Hz because the frequency of human motions is generally less than 10 Hz [MA15].

To create $\widetilde{\theta}_i$, the resultant acceleration of each sensor measurement is calculated by $a_i(k) = \sqrt{x_i(k)^2 + y_i(k)^2 + z_i(k)^2}$. Next, noises are removed by applying a moving average filter to $\{a_i(1), a_i(2), \ldots, \}$. As a result, $\widetilde{a}_i(k) = \frac{1}{H+1} \sum_{n=0}^{H} a(k-n)$ is obtained, where H represents the sliding window length for smoothing adjacent resultant accelerations. Note that the selection of H depends on both the F_{max} and the sampling frequency $1/T_a$.

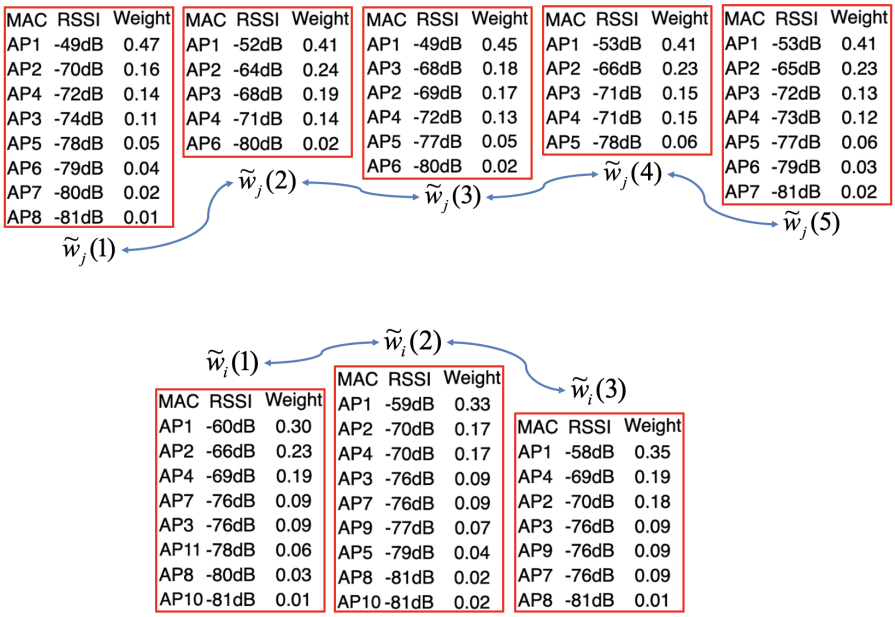
Subsequently, we create the frequency-domain movement signature $\widetilde{\phi}_i$ by applying the FFT to $\widetilde{\theta}_i$. Essentially, the information contained in the frequency-domain $\widetilde{\phi}_i$ is the same as the time-domain $\widetilde{\theta}_i$. The reason of considering the information in the frequency domain lies in, high-frequency and short noises generated by human motions can be easily observed and filtered out. Therefore, we utilize the frequency-domain $\widetilde{\phi}_i$ to compute the similarity between movement signatures.

3.3 Quantification of Cyber Distances

Given two users d_i and d_j, the proposed system respectively quantifies the cyber distances between their mobility signatures and movement signatures. The cyber distance between users' mobility signatures is analyzed from two points of view: *inter-similarity* between their mobility signatures and *self-similarity* along individual



(a) Inter-similarity between d_i and d_j.



(b) Self-similarity for each device.

Figure 3.2: Examples of mobility similarity: (a) inter-similarity between two devices' mobility trajectories, and (b) self-similarity along individual mobility trajectories.

mobility signatures. The inter-similarity quantifies the resemblance between two sequences of mobility signatures along d_i's and d_j's mobility trajectories. The self-similarity evaluates the stability of each individual's mobility and network conditions in the environment along individual mobility trajectory. On the other hand, the cyber distance between their movement signatures is computed to assess the microscopic similarity made by body motions.

3.3.1 Inter-similarity between Mobility Signatures

The key idea of computing inter-similarity between mobility signatures observed by two devices is to analyze their overlapping Wi-Fi APs and associated weights. To this end, a *similarity score* is first defined for each pair of mobility signatures. Next, given two time-series sequences of consecutive mobility signatures observed by d_i and d_j over a time interval I, a DTW-based algorithm is introduced to quantify their inter-similarity based on the RSS correlation. The goal of this algorithm is to match the mobility signatures collected by d_i and d_j such that the accumulated similarity score is maximized.

Definition 1. (Similarity score of a pair of mobility signatures) Given $\widetilde{w}_i(k)$ and $\widetilde{w}_j(k')$, the *similarity score* between the two mobility signatures is defined by

$$S(\widetilde{w}_i(k), \widetilde{w}_j(k')) = \sum_{c_l \in \mathcal{M}(\widetilde{w}_i(k))} \min\{\Upsilon(c_l, \widetilde{w}_i(k)), \Upsilon(c_l, \widetilde{w}_j(k'))\}, \tag{3.3}$$

where $\mathcal{M}(\widetilde{w}_i(k))$ is the set of BSSIDs in the mobility signature $\widetilde{w}_i(k)$. Here, $\Upsilon(c_l, \widetilde{w}_i(k))$ is a function to extract the weight of a given key BSSID c_l from $\widetilde{w}_i(k)$, which is defined as:

$$\Upsilon(c_l, \widetilde{w}_i(k)) = \begin{cases} \sigma_l \text{ , if } (c_l, r_l, \sigma_l) \in \widetilde{w}_i(k); \\ 0 \text{ , otherwise.} \end{cases} \tag{3.4}$$

Fig. 3.2 (a) shows an example of two devices' mobility signatures along their mobility trajectories, where $S(\widetilde{w}_i(2), \widetilde{w}_j(1)) = 0.33+0.16+0.14+0.09+0.02+0+0.04+0.01+0 = 0.79$.

Next, given two time-series sequences of consecutive mobility signatures observed by d_i and d_j during a time interval I, denoted by $\widetilde{\mathbb{W}}_i = \{\widetilde{w}_i(1), \widetilde{w}_i(2), \dots, \}$ and $\widetilde{\mathbb{W}}_j = \{\widetilde{w}_j(1), \widetilde{w}_j(2), \dots, \}$, the *inter-similarity quantification algorithm* is devised to measure the resemblance between them. Here, $\widetilde{\mathbb{W}}_i$ and $\widetilde{\mathbb{W}}_j$ are also referred to as the mobility trajectories of d_i and d_j, respectively. The pseudo code for evaluating the inter-similarity between $\widetilde{\mathbb{W}}_i$ and $\widetilde{\mathbb{W}}_j$ is shown in Algorithm 1.

Initially, a $|\widetilde{\mathbb{W}}_i| \times |\widetilde{\mathbb{W}}_j|$ zero matrix, denoted by D, is initialized for keeping pairwise similarity scores between mobility signatures. In addition, a $|\widetilde{\mathbb{W}}_i| \times |\widetilde{\mathbb{W}}_j|$ zero matrix, denoted by G, is initialized to iteratively search for the *optimal matching* between d_i's mobility signatures and d_j's mobility signatures, where the optimal matching

results in the maximal accumulation of similarity scores along the two sequences $\widetilde{\mathbb{W}}_i$ and $\widetilde{\mathbb{W}}_j$. Then, each entry $D(p,q)$ in the matrix is updated by the similarity score between $\widetilde{w}_i(p)$ and $\widetilde{w}_j(q)$ using Eq. (3.3). Afterwards, all entries in G are recursively updated by examining all possibilities of matching pairs starting with $G(1,1)$ and propagating to all entries in G. Thus, the entries in the first column and the first row in the matrix G are first recursively updated. Here, each entry $G(p,q)$ indicates the maximal accumulation of similarity scores along the best matching path so far. Thus, $G(p,q)$ is updated by choosing the maximal accumulation of similarity scores from the enumerated three possibilities in the neighboring entries and the diagonal entry. Finally, the optimal matching path with the maximal accumulation of similarity scores $G(|\widetilde{\mathbb{W}}_i|,|\widetilde{\mathbb{W}}_j|)$ is found. The maximal accumulation of similarity scores is further normalized. Then, the algorithm outputs the $\Phi(\widetilde{\mathbb{W}}_i,\widetilde{\mathbb{W}}_j)$ that is the inter-similarity between $\widetilde{\mathbb{W}}_i$ and $\widetilde{\mathbb{W}}_j$.

	$\widetilde{w}_j(1)$	$\widetilde{w}_j(2)$	$\widetilde{w}_j(3)$	$\widetilde{w}_j(4)$	$\widetilde{w}_j(5)$
$\widetilde{w}_i(1)$	0.74	0.76	0.69	0.77	0.76
$\widetilde{w}_i(2)$	0.79	0.73	0.76	0.78	0.77
$\widetilde{w}_i(3)$	0.77	0.76	0.74	0.77	0.76

(a) The matrix of similarity scores.

	$\widetilde{w}_j(1)$	$\widetilde{w}_j(2)$	$\widetilde{w}_j(3)$	$\widetilde{w}_j(4)$	$\widetilde{w}_j(5)$
$\widetilde{w}_i(1)$	1.48	2.24	2.93	3.70	4.46
$\widetilde{w}_i(2)$	2.27	3.00	3.76	4.54	5.31
$\widetilde{w}_i(3)$	3.04	3.80	4.54	5.33	6.09

(b) The optimal matching path between mobility signatures.

Figure 3.3: An example of inter-similarity computation.

Fig. 3.3 gives an example of inter-similarity quantification for the two devices in Fig. 3.2(a). The matrix D with the pairwise similarity scores between two sequences of mobility signatures is shown in Fig. 3.3(a). As shown in Fig. 3.3(b), initially, $G(1,1) = 2 \times S(\widetilde{w}_i(1),\widetilde{w}_j(1)) = 2 \times 0.74$. In this example, numbers in all entries are rounded to two decimal places. After the first column and the first row are updated, $G(2,2) = \max\{2.24 + 0.73, 2.27 + 0.73, 1.48 + 2 \times 0.73\} = 2.27 + 0.73 = 3.00$ is recursively updated. Similarly, all entries in the matrix G are updated, where each entry maintains the maximal accumulation of similarity scores so far among the three possibilities. So, the optimal matching path (shown in bold) is found by backtracking from the maximal accumulation of similarity scores $G(3,5) = 6.09$ to $G(1,1) = 1.48$. The corresponding matching between mobility signatures of the two devices along the optimal matching path is shown with double arrows in Fig. 3.2(a), where $\widetilde{w}_j(1)$ is matched with $\widetilde{w}_i(1)$ and $\widetilde{w}_i(2)$, and $\widetilde{w}_j(2)$ is only matched with $\widetilde{w}_i(2)$. Finally, the inter-similarity is $\Phi(\widetilde{\mathbb{W}}_i,\widetilde{\mathbb{W}}_j) = \frac{6.09}{(3+5)}$.

Compared with other trajectory matching methods, such as Levenshtein distance, Fréchet distance, and Euclidean distance [DL18], the DTW-based Algorithm 1 can more effectively match asynchronously collected sequences. This is because the mobility trajectories $\widetilde{\mathbb{W}}_i$ and $\widetilde{\mathbb{W}}_j$ may be collected at different timestamps

Algorithm 1: Inter-similarity Quantification $(\widetilde{\mathbb{W}}_i, \widetilde{\mathbb{W}}_j)$.

Input : $\widetilde{\mathbb{W}}_i$: Mobility signatures collected by device d_i;
$\widetilde{\mathbb{W}}_j$: Mobility signatures collected by device d_j;
Output : $\Phi(\widetilde{\mathbb{W}}_i, \widetilde{\mathbb{W}}_j)$

//**Initialization:**
// A $|\widetilde{\mathbb{W}}_i| \times |\widetilde{\mathbb{W}}_j|$ matrix is initialized for keeping pairwise similarity scores.
$D \leftarrow 0_{|\widetilde{\mathbb{W}}_i|,|\widetilde{\mathbb{W}}_j|}$
// A $|\widetilde{\mathbb{W}}_i| \times |\widetilde{\mathbb{W}}_j|$ matrix is initialized for tracking the best matching path.
$G \leftarrow 0_{|\widetilde{\mathbb{W}}_i|,|\widetilde{\mathbb{W}}_j|}$;
//**Compute similarity scores:**
for *p from* 1 *to* $|\widetilde{\mathbb{W}}_i|$ **do**
  for *q from* 1 *to* $|\widetilde{\mathbb{W}}_j|$ **do**
    $D(p,q) \leftarrow S(\widetilde{w}_i(p), \widetilde{w}_j(q))$ // using Eq. (3.3).
//**Update optimal match between two time-series sequences:**
//Start the calculation with $G(1,1)$.
$G(1,1) \leftarrow 2 \times S(\widetilde{w}_i(1), \widetilde{w}_j(1))$;
// Calculate the first column.
for *p from 2 to* $|\widetilde{\mathbb{W}}_i|$ **do**
  $G(p,1) \leftarrow G(p-1,1) + S(\widetilde{w}_i(p), \widetilde{w}_j(1))$;
// Calculate the first row.
for *q from 2 to* $|\widetilde{\mathbb{W}}_j|$ **do**
  $G(1,q) \leftarrow G(1,q-1) + S(\widetilde{w}_i(1), \widetilde{w}_j(q))$;
for *p from 2 to* $|\widetilde{\mathbb{W}}_i|$ **do**
  for *q from 2 to* $|\widetilde{\mathbb{W}}_j|$ **do**
    $vertical \leftarrow G(p-1,q) + S(\widetilde{w}_i(p), \widetilde{w}_j(q))$;
    $horizontal \leftarrow G(p,q-1) + S(\widetilde{w}_i(p), \widetilde{w}_j(q))$;
    $diagonal \leftarrow G(p-1,q-1) + 2 \times S(\widetilde{w}_i(p), \widetilde{w}_j(q))$;
    $G(p,q) \leftarrow \max\{vertical, horizontal, diagonal\}$;
//**Normalization:** $\Phi(\widetilde{\mathbb{W}}_i, \widetilde{\mathbb{W}}_j) \leftarrow G(|\widetilde{\mathbb{W}}_i|, |\widetilde{\mathbb{W}}_j|)/(|\widetilde{\mathbb{W}}_i| + |\widetilde{\mathbb{W}}_j|)$;
return $\Phi(\widetilde{\mathbb{W}}_i, \widetilde{\mathbb{W}}_j)$;

and sampling rates, d_i's sensing period may not align with d_j's sensing period during interval I. The proposed Algorithm 1 is designed based on DTW, which nonlinearly warps the time axes of $\widetilde{\mathbb{W}}_i$ and $\widetilde{\mathbb{W}}_j$ to find their optimal alignment, thereby capturing the actual temporal relations between the two mobility trajectories despite asynchronous sampling. In contrast, the Levenshtein distance matches trajectories through edit operations, but it cannot capture the temporal correlations between mobility signatures. The Fréchet distance matches trajectories based on geometric similarity, yet modeling geometric relations between mobility signatures is difficult because each signature contains multi-dimensional information such as BSSID, RSSI, and AP weights. The Euclidean distance performs pointwise comparisons, but

even subtle temporal misalignment can significantly distort the computed distance, leading to unreliable matching results.

3.3.2 Self-Similarity of Mobility Signatures

Since movement signatures on each individual's mobility trajectory are affected by moving speed and ambient network conditions, they can provide insights into an individual's mobility stability and spatial features. Therefore, the self-similarity of an individual's mobility signature is quantified as the average similarity score of the observed sequence of mobility signatures observed by the device.

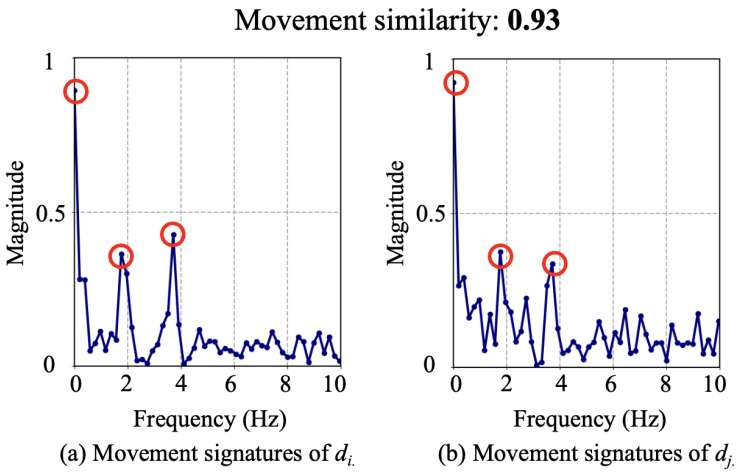
Definition 2. (Self-similarity of individual mobility signatures) Given a sequence of mobility signatures for d_i, $\widetilde{\mathbb{W}}_i = \{\widetilde{w}_i(1), \widetilde{w}_i(2), \dots, \}$, the *self-similarity* of $\widetilde{\mathbb{W}}_i$ is defined by

$$\Omega(\widetilde{\mathbb{W}}_i) = \frac{\sum_{k=1}^{|\widetilde{\mathbb{W}}_i|-1} S(\widetilde{w}_i(k), \widetilde{w}_i(k+1))}{|\widetilde{\mathbb{W}}_i| - 1}. \tag{3.5}$$

3.3.3 Similarity between Movement Signatures

Definition 3. (Movement similarity) Given two movement signatures $\widetilde{\mathbb{A}}_i = (\widetilde{\theta}_i, \widetilde{\phi}_i)$ and $\widetilde{\mathbb{A}}_j = (\widetilde{\theta}_j, \widetilde{\phi}_j)$ of d_i and d_j, the movement similarity between them is defined based on the normalized root-mean-square-distance between $\widetilde{\phi}_i$ and $\widetilde{\phi}_j$:

$$\Psi_f(\widetilde{\phi}_i, \widetilde{\phi}_j) = 1 - \frac{\sqrt{\sum_{p=1}^{L} (\widetilde{b}_i(p) - \widetilde{b}_j(p))^2 / L}}{F_{max}}. \tag{3.6}$$



(a) Movement signatures of d_i. (b) Movement signatures of d_j.

Figure 3.4: Frequency-domain movement signatures of two devices when they are making movements together.

Here, $L = |\widetilde{\phi}_i| = |\widetilde{\phi}_j|$ is the length of converted accelerations over the frequency spectrum, and F_{max} is the maximal frequency of interest depending on the frequency

of human motions. Fig. 3.4 demonstrates that the movement signatures of d_i and d_j over the frequency spectrum are highly similar to each other when they are making movements together. The movement similarity between them is $\Psi_f(\widetilde{\phi}_i, \widetilde{\phi}_j) = 0.93$.

3.4 Cyber-physical Social Distancing

3.4.1 Detection Algorithm

The detection algorithm is devised to determine whether or not two devices are physically non-separate based on the mobility similarity obtained from Wi-Fi data and the movement similarity derived from accelerations. Given $(\mathbb{W}_i, \mathbb{A}_i)$ and $(\mathbb{W}_j, \mathbb{A}_j)$ collected by d_i and d_j, their mobility signatures, $\widetilde{\mathbb{W}}_i$ and $\widetilde{\mathbb{W}}_j$, and movement signatures, $\widetilde{\mathbb{A}}_i$ and $\widetilde{\mathbb{A}}_j$, can be created using the proposed approaches in Section 3.2. Then, their mobility similarity is computed from two perspectives: inter-similarity $\Phi(\widetilde{\mathbb{W}}_i, \widetilde{\mathbb{W}}_j)$ using Algorithm 1 and their self-similarities $\Omega(\widetilde{\mathbb{W}}_i)$ and $\Omega(\widetilde{\mathbb{W}}_j)$ using Eq. (3.5). In addition, their movement similarity $\Psi_f(\widetilde{\phi}_i, \widetilde{\phi}_j)$ is computed using Eq. (3.6). Therefore, the inputs of the detection algorithm including their inter-similarity, self-similarities, and movement similarity are previously calculated for making a binary decision on the relationship of the two devices. Let $\mathbb{V}(d_i, d_j)$ denote the result decided by the algorithm, where $\mathbb{V}(d_i, d_j) =$ "$separate$" indicates a physically separate status ("separate" for short), and $\mathbb{V}(d_i, d_j) = non-separate$ indicates a physically non-separate status ("non-separate" for short). Initially, $\mathbb{V}(d_i, d_j) = separate$.

The sensor-based movement similarity is first examined to trigger the following examination of Wi-Fi-based similarity metrics. If $\Psi_f(\widetilde{\phi}_i, \widetilde{\phi}_j) > \tau_f$, then their self-similarities are further checked to examine the mobility stability and spatial characteristics. Otherwise, the algorithm terminates. Here, the τ_f is a predefined threshold of movement similarity. Since a higher movement similarity is most likely resulting from close physical contact, the algorithm further checks the ratio of $\frac{\min\{\Omega(\widetilde{\mathbb{W}}_i), \Omega(\widetilde{\mathbb{W}}_j)\}}{\max\{\Omega(\widetilde{\mathbb{W}}_i), \Omega(\widetilde{\mathbb{W}}_j)\}}$ based on their self-similarities. If $\frac{\min\{\Omega(\widetilde{\mathbb{W}}_i), \Omega(\widetilde{\mathbb{W}}_j)\}}{\max\{\Omega(\widetilde{\mathbb{W}}_i), \Omega(\widetilde{\mathbb{W}}_j)\}} > \tau_s$, then the inter-similarity between them is further examined. In this case, it implies that the two devices have similar mobility stability or spatial characteristics resulting from similar network conditions in their environments. Otherwise, the algorithm terminates. Here, τ_s is a predefined threshold for self-similarity. When the algorithm further examines their inter-similarity, if $\Phi(\widetilde{\mathbb{W}}_i, \widetilde{\mathbb{W}}_j) > \tau_i$, then $\mathbb{V}(d_i, d_j) = non-separate$ is set. Otherwise, the algorithm terminates with the initial $\mathbb{V}(d_i, d_j) = separate$.

3.4.2 Complexity of the Detection Algorithm

Given $(\mathbb{W}_i, \mathbb{A}_i)$ and $(\mathbb{W}_j, \mathbb{A}_j)$ collected by two devices d_i and d_j, the computational complexity of the detection algorithm is analyzed in this section. We denote by $M = \max\{|\mathbb{W}_i|, |\mathbb{W}_j|\}$ the maximal number of Wi-Fi measurements collected by a device within a certain time interval, and denote by N the maximal number of

Wi-Fi APs observed during a scan. We denote by J the number of accelerations within a certain time interval.

Time complexity of signature creation: For each Wi-Fi scan, the RSS of observed Wi-Fi APs are first sorted in $O(N \log N)$. Next, assigning non-negative weights takes $O(N)$ by checking the entire list. Therefore, creating a mobility signature (in Section 3.2.1) takes $O(N \log N + N) = O(N \log N)$. Given Wi-Fi measurements collected by two devices d_i and d_j during a time interval, which takes $O((N \log N) \times 2M) = O(MN \log N)$ to create their corresponding signatures $\widetilde{\mathbb{W}}_i$ and $\widetilde{\mathbb{W}}_j$. On the other hand, given $\mathbb{A}_i$, creating the time-domain movement signature $\widetilde{\theta}_i$ requires $O(J)$ for computing the resultant accelerations and the subsequent filtering, and it further takes $O(J \log J)$ for the FFT to convert the time-domain movement signature to the frequency-domain movement signature $\widetilde{\phi}_i$. So, creating a movement signature $\widetilde{\mathbb{A}}_i = (\widetilde{\theta}_i, \widetilde{\phi}_i)$ (in Section 3.2.2) takes $O(J) + O(J \log J) = O(J \log J)$. It totally takes $O(2 \times J \log J) = O(J \log J)$ to create the movement signatures $\widetilde{\mathbb{A}}_i$ and $\widetilde{\mathbb{A}}_j$ for two devices. As a result, the overall complexity for creating the mobility signatures and movement signatures is $O(MN \log N) + O(J \log J)$.

Time complexity of cyber distance computation: Initially, computing the inter-similarity based on Algorithm 1 takes $O(M^2N^2)$, where $O(M^2N^2)$ is required to compute pairwise similarity scores and $O(M^2)$ is necessary for updating the optimal match between two time-series sequences. Here, computing a similarity score for each pair of mobility signatures based on Eq. (3.3) takes $O(N^2)$ to cross-check all possible combinations of Wi-Fi APs in the two mobility signatures, and there are totally $O(M^2)$ pairs to compute their similarity scores. Next, $O(M^2)$ combinations between the two devices' mobility signatures are checked to find the optimal matching path. Subsequently, the computation of the self-similarity for each device (in Section 3.3.2) takes $O(MN^2)$ including $O(N^2)$ for computing a similarity score of any two consecutive mobility signatures of this device and $O(M)$ for accumulating them by examining the entire list. For two devices, it requires $O(2MN^2) = O(MN^2)$ to compute their self-similarities. Consequently, the overall complexity for computing the mobility similarity is $O(M^2N^2) + O(MN^2) = O(M^2N^2)$. On the other hand, the complexity for computing movement similarity between the two devices based on Eq. (3.6) is $O(J)$. Overall, it costs $O(M^2N^2 + J)$ for cyber distance computation.

Time complexity of the detection algorithm: As all inputs to the detection algorithm have been computed (in Algorithm 1, Section 3.3.2, and Section 3.3.3), the time complexity is already included in the previous analyses. Therefore, the detection algorithm only takes $O(1)$ to check the mobility similarity and movement similarity for making a decision based on the pre-defined threshold.

Total complexity: Overall, creating mobility and movement signatures takes $O(MN \log N) + O(J \log J)$ in total, computing cyber distances takes $O(M^2N^2 + J)$, and decision-making takes only $O(1)$. So, its cost is $O(MN \log N) + O(J \log J) + O(M^2N^2 + J) + O(1) = O(M^2N^2 + J \log J)$. In general, M is limited by the fixed time interval I and Wi-Fi scanning frequency, N is influenced by the communication range of the Wi-Fi APs and the network condition in the environment, and J is

determined by the sampling frequency of the acceleration. The proposed system set I to 6 seconds and the scanning frequency not greater than 1 Hz. Therefore, M is limited to a very small number. In addition, N is also a small number since the number of Wi-Fi APs in an environment even with dense networks is roughly less than 25 based on our observations in experiments. The sampling frequency of accelerations has to be greater than $2F_{max}$ = 20 Hz (i.e., twice the highest frequency of human motions) to eliminate aliasing according to the Nyquist Shannon sampling theorem. Therefore, J = 120 is the lower bound for a time interval of 6 seconds. The value of J is a small number in our system (with a sampling frequency of 50 Hz). Therefore, real-time applications can be supported in our proposed system because of the low computational complexity.

3.5 Implementation and Demonstration

This work implements a mobile application for data collection, where the Wi-Fi scanning period is set to T_w = 1 second and the acceleration sampling period is set to T_a = 1/50 second. The time interval considered by the detection algorithm is I = 6 seconds. The maximum frequency of human motion is considered to be F_{max} = 10 Hz. Based on the cut-off frequency F_{max} and the sampling frequency $1/T_a$, the default value of H for the moving average filter in Section 3.2.2 and the default value of L for the calculation of the movement similarity in Eq. (3.6) are selected according to the principles in [You01]. Therefore, H = 22 and L = 50 are set. The thresholds τ_f = 0.76, τ_s = 0.4, and τ_i = 0.6 in the detection algorithm are set empirically.

The user interface of the proposed system is presented in Fig. 3.5, where the result of the real-time detection and its corresponding mobility similarity and movement similarity between two users are demonstrated. When the two users are physically non-separate, a warning alarm is given. This snapshot shows the result when the two users are keeping static in two different rooms. As shown in this figure, a conclusion is given by the system that they are "physically separate" due to a small inter-similarity. As the two users are static, the values of their self-similarity are similar to each other and very stable. The movement similarity is large also because of the static status.

3.6 Experiments and Evaluation

In this section, we conduct extensive experiments in two small-scale environments and a large-scale environment. In addition, comprehensive 2D and 3D mobility datasets are both utilized to provide in-depth analysis. In small-scale environments, the Wi-Fi network conditions (i.e., detectable Wi-Fi APs) along trajectories does not change too much because users' trajectories are limited by the sizes of the environments. In the large-scale environment, users move along longer trajectories,

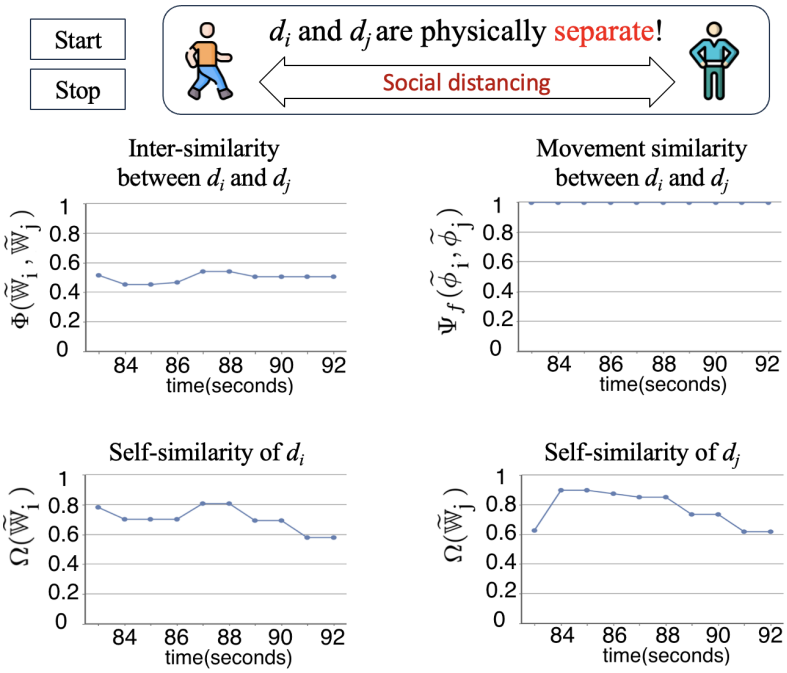


Figure 3.5: The user interface of the proposed system.

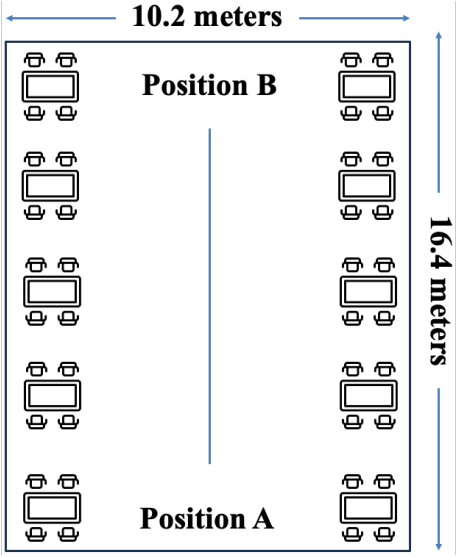
through sparse and dense networks, with significant spatial transitions. In small-scale environments, two devices are placed in the same person's backpack to verify a more restrictive non-separate scenario, where they actually move together on the same mobile entity. Next, we conduct experiments in the large environment to verify the robustness of our design, which is not affected by the wearing position. Furthermore, multiple users change their relationships from separate to non-separate and device carrying positions while taking diverse trajectories between different areas. In this case, our mobility similarity metrics is first compared against the metric proposed in [HS19]. Afterwards, we analyze timeliness in indicating proximity levels, combinations of similarity metrics, parameter selection, and cost of energy and communications through comprehensive experiments in the large-scale environment. Finally, extensive mobility datasets are utilized to conduct statistical analysis of synchronicity between cyber and physical distances.

3.6.1 Small-scale Environments and Mobility Scenarios

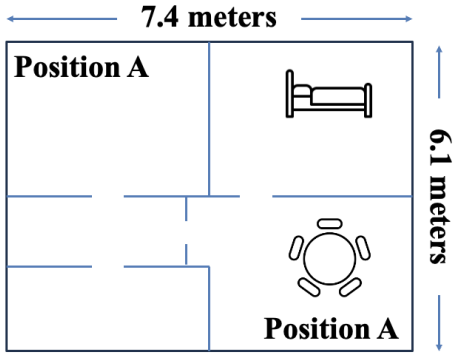
First, Fig. 3.6 presents two indoor environments with completely different Wi-Fi network conditions, i.e., the faculty library and the private apartment, where the preliminary experiments are conducted. In the faculty library, more than 15 Wi-Fi APs are detectable. In the private apartment, only less than 10 Wi-Fi APs can be observed. Moreover, due to the higher crowd density and more dynamic human

(a) Faculty library.

(b) Private apartment.



(c) Floor plan of library.



(d) Floor plan of apartment.

Figure 3.6: Two small-scale experiments.

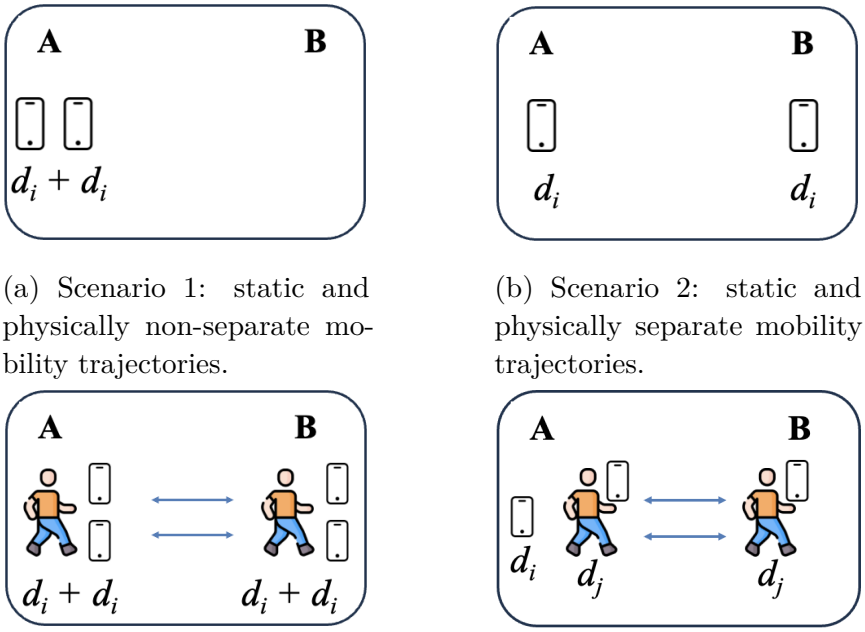
mobility, the Wi-Fi signal in the university library changes more significantly than in the private apartment.

In the two environment, two mobile devices simulate four mobility scenarios to for the experiments, as shown in Fig. 3.7. We pre-select positions A and B, which are located in opposite corners and a long way apart in both environments. The two devices can therefore move along a straight line in the university library or along a meandering path through several small rooms in the private apartment. The two devices simulate each of the following scenarios shown in Fig. 3.7.

(1) Scenario 1: Two devices, d_i and d_j, are placed side by side at Position A statically. In this case, the two devices are considered as non-separate.
(2) Scenario 2: Device d_i is statically placed at Position A, while device d_j is statically placed at Position B. In this case, the two devices are considered as separate.
(3) Scenario 3: The two devices d_i and d_j are carried by the same person, moving back and forth between Position A and Position B. In this case, the two devices are considered as non-separate.

(4) Scenario 4: Device d_i is stationary at Position A, while device d_j is carried by a person moving back and forth between Position A and Position B. In this case, the two devices are occasionally separate and occasionally non-separate.

The detection algorithm is continuously executed for 200 seconds in each scenario. The detection results are compared with the ground truth in the four scenarios to preliminarily analyze the performance of our system.



(a) Scenario 1: static and physically non-separate mobility trajectories.

(b) Scenario 2: static and physically separate mobility trajectories.

(c) Scenario 3: dynamic and physically non-separate mobility trajectories.

(d) Scenario 4: dynamic and hybrid mobility trajectories.

Figure 3.7: Three mobility scenarios of the two devices.

3.6.2 Performance Metrics

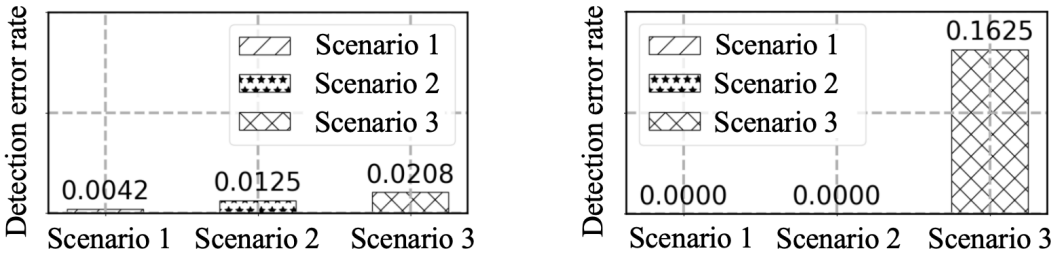
The following metrics are proposed to evaluate the performance of the proposed system. A true positive result refers to two non-separate devices being detected as non-separate, while a true negative result refers to two separate devices being detected as separate. Similarly, a false positive result refers to two separate devices being detected as non-separate, whereas a false negative result refers to two non-separate devices being detected as separate. We denote by TP, TN, FP, and FN the number of true positive results, the number of true negative results, the number of false positive results, and the number of false negative results respectively. Therefore, the sensitivity, specificity, and accuracy can be defined as follows: $sensitivity = \frac{TP}{TP+FN}$, $specificity = \frac{TN}{TN+FP}$, and $accuracy = \frac{TP+TN}{TP+TN+FP+FN}$. Sensitivity is the ability to

correctly identify two non-separate devices. Specificity is the ability to correctly identify two separate devices. Accuracy is the ability to correctly identify the relationship between two devices.

To further study the impact of different mobility trajectories on system performance, the detection error rate for each mobility scenario is defined as the ratio of the number of incorrect detection results to the total number of tests in each scenario.

3.6.3 Effects of Mobility Trajectories

The experimental results of the detection error rates in the four mobility scenarios are shown in Fig. 3.8. When two devices are static in Scenario 1 and Scenario 2, the detection error rates are very low in both two environments. Fig. 3.8(a) shows that when the two devices dynamically take mobility trajectories in the library, the detection error rate slightly increases in Scenario 3. A larger increase in the detection error rate for Scenario 3 in the private apartment is presented in Fig. 3.8(b), because the person moving along a winding route through several small rooms may cause greater interference to the propagation of the Wi-Fi signal in such a small apartment. The influence of human mobility on signal propagation also can be seen in static scenarios in Fig. 3.8(b), where sparser crowds and less dynamic human mobility in the apartment result in lower detection error rates in comparison to static scenarios in the library.



(a) Results in the library.

(b) Results in the apartment.

Figure 3.8: Detection error rates with different mobility trajectories.

3.6.4 Analyses of Device-to-Device Cyber Distances

In this part, we further explore the advantages of the proposed cyber distances and analyze how these advantages facilitate a more reliable detection. First, the device-to-device (D2D) mobility similarity and movement similarity in the faculty library with denser networks are investigated. Next, the performance in the two extremely different environments in terms of network density, crowd sizes, mobility dynamics, and space sizes are compared. Finally, the effectiveness and importance of

the proposed movement similarity is also analyzed when the two users have physical contact, i.e., social handshakes.

Effectiveness of D2D Cyber Distances

First, the mobility similarity between the two devices from inter-similarity and self-similarity perspectives is explored. Next, we study the movement similarity between them.

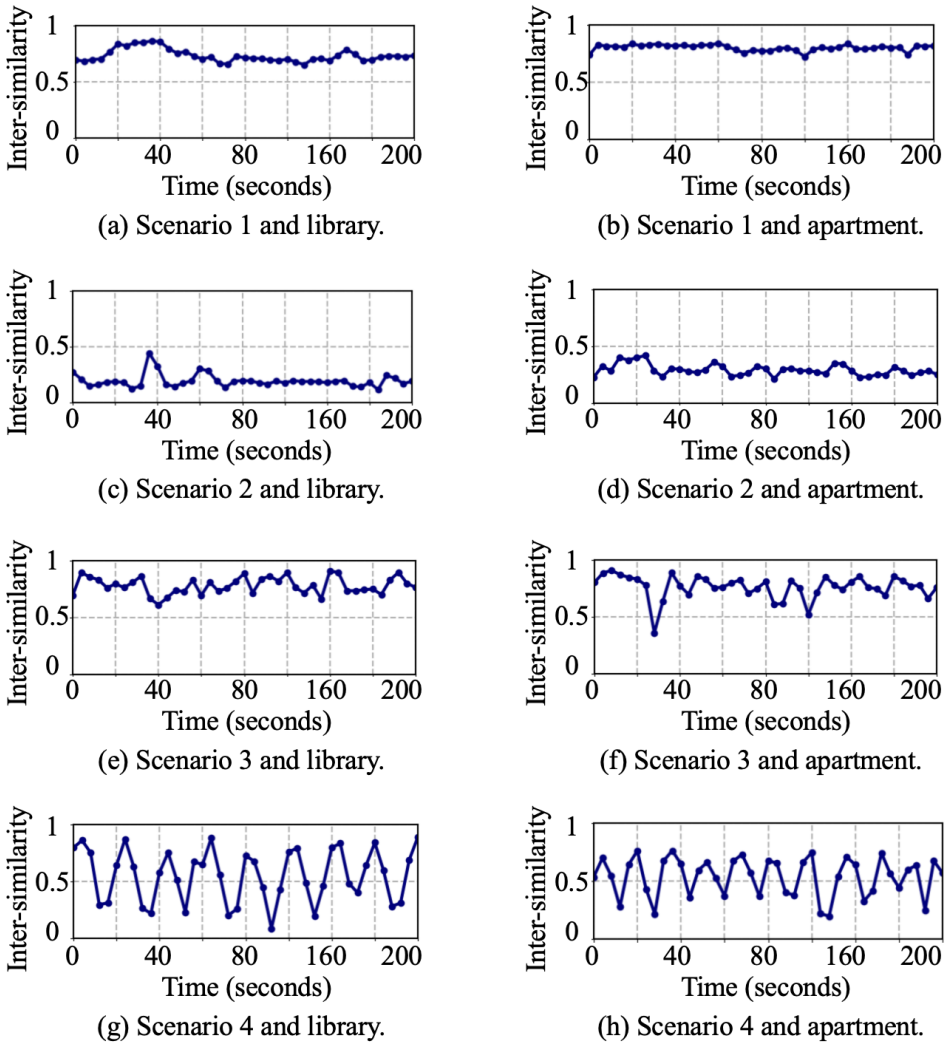


Figure 3.9: Inter-similarity between d_i and d_j in the library and apartment.

The inter-similarities between d_i and d_j in Scenario 1, Scenario 2, Scenario 3, and Scenario 4 in two small-scale environments are shown in Fig. 3.9. In the non-separate scenarios (e.g., Scenarios 1 and Scenarios 3), the inter-similarity between d_i and d_j is significantly larger than the inter-similarity in the separate scenario, i.e., Scenario 2. As can be seen, the proposed inter-similarity offers informative insights for

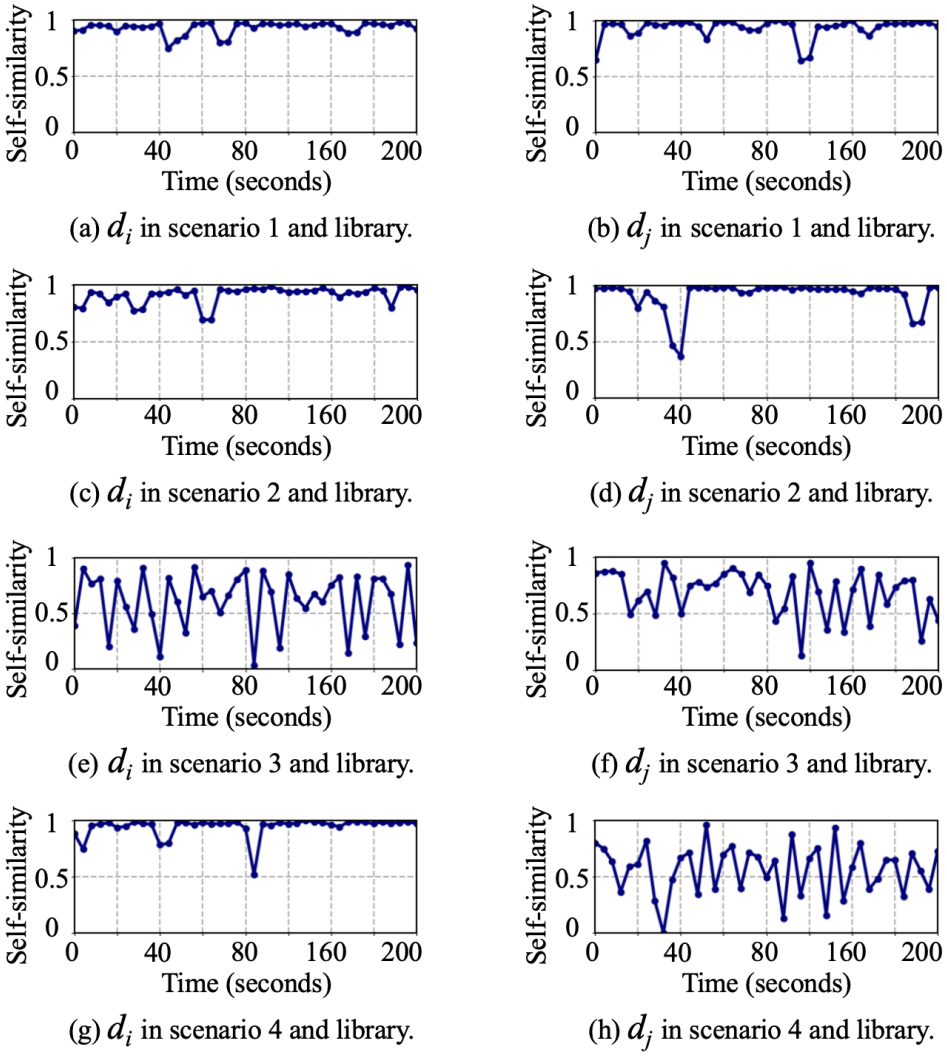
distinguishing between non-separate and separate correlations between devices, even in dynamic Scenario 3. More interesting things occur in Scenario 4, where d_i moves back and forth between two locations while d_j stays in the same location all the time. Fig. 3.9(g) and Fig. 3.9(h) vividly characterize this scenario through inter-similarity. When d_i is moving away from device d_j, their inter-similarity becomes low, when d_i is moving closer to d_j, their inter-similarity immediately increases.

The individual self-similarity is further investigated on each device's mobility stability. Fig. 3.10(a) to Fig. 3.10(d) presents the self-similarity of each device in static scenarios (e.g., Scenario 1 and Scenario 2), since all devices are statically placed, their self-similarities are more stable compared to the results in the dynamic Scenario 3, as shown in Fig. 3.10(e) and Fig. 3.10(f). In Scenario 4, the self-similarity of one device is relatively stable while the self-similarity of the other device fluctuates, because one device is always moving while the other device is always stationary, as shown in Fig. 3.10(g) and Fig. 3.10(h). The proposed self-similarity explicitly reveals the changes of mobility signatures with the moving speeds of devices. Specifically, this metric helps understand the spatial transitions taken by each individual device.

Finally, the movement similarity between the two devices is explored. As can be seen in Fig. 3.11(a), because d_i and d_j are static and non-separate in Scenario 1, the movement similarity between them is $\Psi_f(\widetilde{\phi}_i, \widetilde{\phi}_j) = 1$. Similar result can be found in Fig. 3.11(b) when the two devices are static and separate in Scenario 2. As shown in Fig. 3.11(c), when the two devices are non-separate and moving together in Scenario 3, their movement similarity is very high. The movement similarity in Scenario 4 exhibits a similar pattern to that in Scenario 3, except that the fluctuations in Scenario 4 are slightly larger, as shown in Fig. 3.11(d). This is because the movement statuses of d_i and d_j differ more in Scenario 4.

D2D Cyber Distances in Different Environments

Next, the impact of different environments on the performance of the proposed system is further investigated. Because the experimental results of mobility similarity and movement similarity in the library and in the apartment are similar, the inter-similarities in the two environments are presented that show a significant impact of environmental conditions. Compared to the static scenarios in the library shown in Fig. 3.9(a) and Fig. 3.9(c), the inter-similarities in static scenarios in the apartment are smoother as shown in Fig. 3.9(b) and Fig. 3.9(d). This is because less human mobility in the apartment creates less noise in the environment, leading to better performance in static Scenario 1 and Scenario 2, as shown in Table 3.1. When the two devices are moving together passing through several rooms in the smaller apartment, Fig. 3.9(f) shows stronger variations in inter-similarity in comparison to Fig. 3.9(e) in the library. This is because the size of the apartment is smaller and the moving trajectory is not a simple straight line. Signal fading in the apartment is relatively severe, leading to the worse performance of 83.75% in the dynamic Scenario 3.



(a) d_i in scenario 1 and library. (b) d_j in scenario 1 and library.

(c) d_i in scenario 2 and library. (d) d_j in scenario 2 and library.

(e) d_i in scenario 3 and library. (f) d_j in scenario 3 and library.

(g) d_i in scenario 4 and library. (h) d_j in scenario 4 and library.

Figure 3.10: Self-similarity of each device in the library.

D2D Cyber Distances with Social Handshakes

Further experiments are conducted to investigate how the movement similarity between two users reflects the social handshake behavior. In this experiment, the two users wear the device on their wrists and shake hands for a certain period of time. As can be seen in Fig. 3.12, the movement similarity in a handshaking case is significantly higher than that in a non-handshaking case. It can be concluded that the proposed metrics are capable of measuring the differences in not only macroscopic mobility along moving trajectories but also the microscopic movements caused by physical contact.

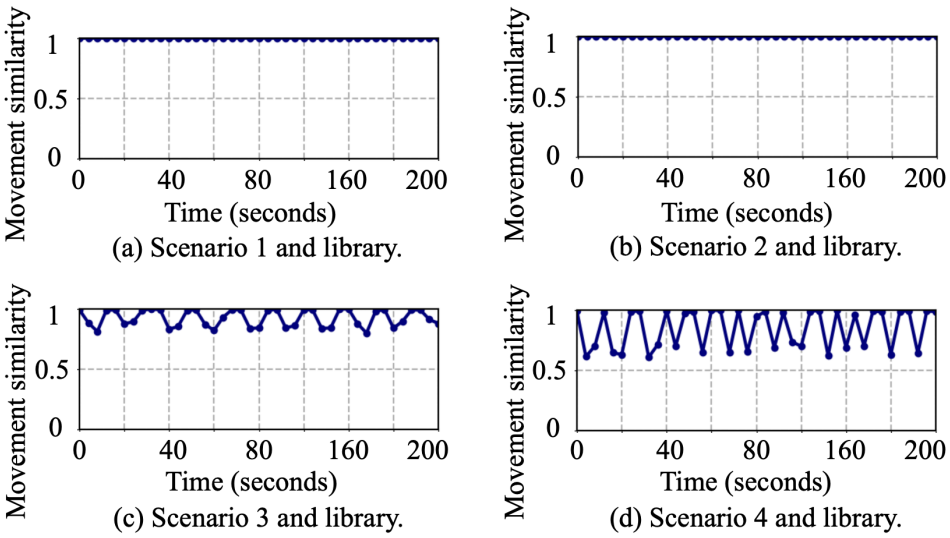


Figure 3.11: Movement similarity between d_i and d_j in the library.

Table 3.1: Performance comparison in two small-scale environments.

	Sensitivity		**Specificity**	**Accuracy**
	Scenario 1	*Scenario 3*	*Scenario 2*	
Library	99.58%	97.92%	98.75%	98.75%
Apartment	100.00%	83.75%	100.00%	94.45%

3.6.5 Advanced Study in a Large-Scale Environment

Setup with Multiple Devices and Diverse Trajectories

In this section, advanced experiments involving multiple devices and diverse trajectories in a large-scale environment are conducted to further validate the scalability and flexibility of the proposed metrics and our system. As shown in Fig. 3.13, the environment includes several office rooms on the left-hand side and student labs on the right-hand side, collectively accommodating more than 20 people. The crowd density and Wi-Fi network deployment on the left-hand side are relatively sparse, whereas the right-hand side experiences more human mobility and a higher density of Wi-Fi networks due to room usage. Consequently, the RSS values on the left-hand side are typically weaker than -55 dB, while those on the right-hand side are generally stronger than -35 dB.

Compared with the experiments in the small-scale environments, the advanced experiments involve more diverse mobility trajectories that traverse areas with varying crowd densities and Wi-Fi network deployments. One-hour statistics collected in these multi-storey environments show that the average numbers of detectable Wi-Fi APs are 10.3 in the private apartment, 17.4 in the library, 8.1 in the left corridor of

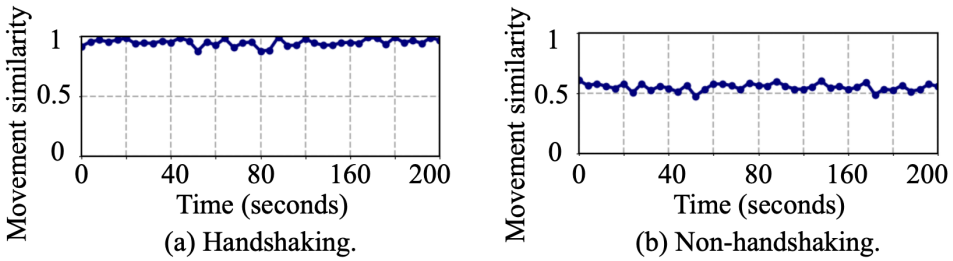


Figure 3.12: The movement similarities for handshaking and non-handshaking scenarios.

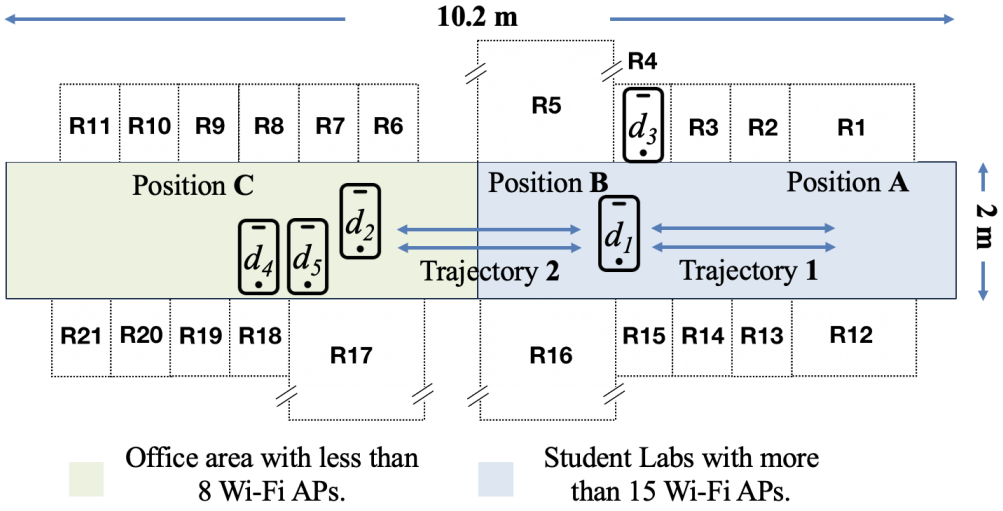
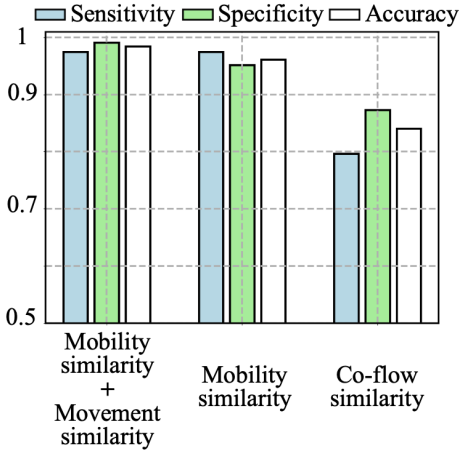


Figure 3.13: The floor plan of the large-scale environment.

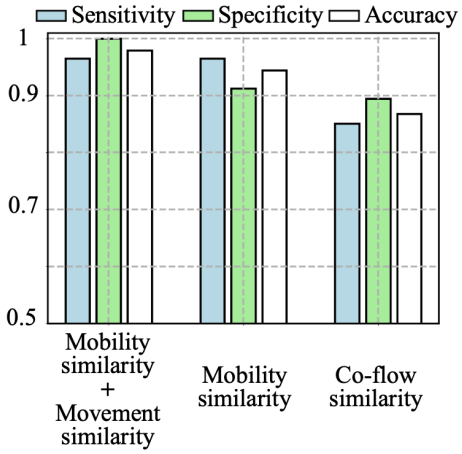
the large-scale environment, and 18.8 in the right corridor. Five different brands of devices are used to verify the flexibility of our method. Three positions (A, B, and C), shown in Fig. 3.13, are selected to construct diverse trajectories and are located in front of Room R1, Room R4, and Room R7, respectively, with 10 meters between A and B and between B and C. Two round-trip trajectories are considered: Trajectory 1 between A and B, and Trajectory 2 between B and C. When users arrive at any start or destination position, they stop for a few seconds to record the ground truth positions via the mobile user interface and then continue moving. The entire experiment lasts 1500 seconds. During the first 900 seconds, devices d_1 and d_2 are held by two users walking along trajectories 1 and 2, respectively, while swinging their arms naturally. After 900 seconds, the two users meet at Position B and device d_2 is handed to the user carrying d_1, after which both devices are placed in the user's palms and move together along Trajectory 1 until the end of the experiment. Throughout the experiment, devices d_4 and d_5 remain static at Position C, and device d_3 remains static at Position B.

Performance Comparisons in Complex Scenarios

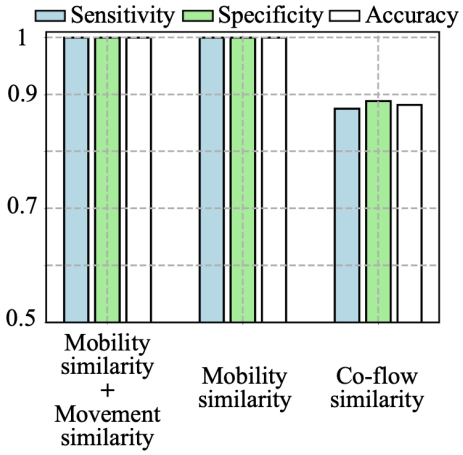
We compare the proposed similarity metrics against the co-flow similarity proposed in [HS19] (denoted as "co-flow"), which combines the Tanimoto similarity, Adamic-Adar similarity, and DTW-based similarity for group detection using Wi-Fi data. Fig. 3.14 presents both the overall detection performance and a comprehensive analysis across multiple mobility scenarios. The dynamic scenario refers to cases where both devices in a pair are moving, including separate and non-separate situations. The static scenario refers to cases where both devices remain stationary. The hybrid scenario refers to cases where one device is moving while the other is static.



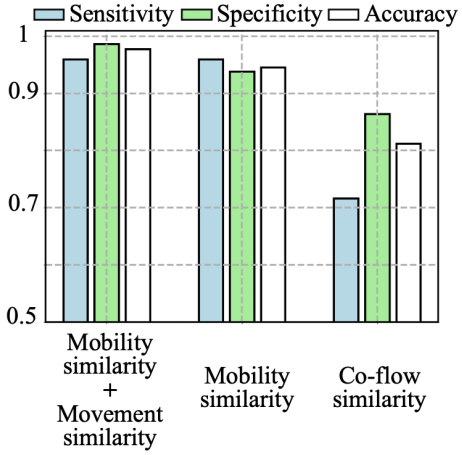
(a) Overall performance.



(b) The dynamic scenario.



(c) The static scenario.



(d) The hybrid scenario.

Figure 3.14: Comprehensive analyses of our similarity metrics vs. co-flow similarity metrics.

As shown in Fig. 3.14(a), when both the Wi-Fi-based mobility similarity and the sensor-based movement similarity are used, our approach achieves consistently high

sensitivity, specificity, and accuracy across diverse and complex mobility scenarios. The average sensitivity of 97.45% indicates that our approach correctly identifies non-separate cases in 97.45% of the instances where the two devices are indeed non-separate. The average specificity of 99.13% indicates that separate cases are correctly identified in 99.13% of the instances where the devices are actually separate. The average accuracy of 98.41% further demonstrates that our approach remains robust against variations in environmental conditions and human mobility. When devices are moving (i.e., in the dynamic and hybrid scenarios), the performance slightly decreases. As shown in Fig. 3.14(b) and Fig. 3.14(d), the dynamic scenario yields a sensitivity of 96.55%, a specificity of 100.00%, and an accuracy of 97.92%, while the hybrid scenario yields a sensitivity of 96.02%, a specificity of 98.67%, and an accuracy of 97.74%. By contrast, Fig. 3.14(c) shows that in the static scenario, our approach correctly identifies all non-separate and separate cases, owing to the reduced variability in both network conditions and human movement.

When only the Wi-Fi-based mobility similarity is used, Fig. 3.14(a) shows that our approach still outperforms co-flow, where the sensitivity, specificity, and accuracy of co-flow are 79.63%, 87.33%, and 84.03%, respectively. The specificity and accuracy of our method decrease slightly to 95.14% and 96.13%, while the sensitivity remains unchanged. This is because Wi-Fi data alone cannot reliably distinguish some separate cases in dynamic and hybrid scenarios, where device states fluctuate due to small-scale body motions or changes in carrying positions. Evidence of this can be seen in Fig. 3.14(b) and Fig. 3.14(d), where the specificity and accuracy drop to 91.23% and 94.44% in the dynamic scenario and to 93.87% and 94.62% in the hybrid scenario when only Wi-Fi-based mobility similarity is used. In contrast, the performance in the static scenario remains unaffected, as shown in Fig. 3.14(c), indicating that movement similarity primarily benefits non-static cases by capturing subtle changes of device states that Wi-Fi data alone cannot, while offering no additional value when all devices remain still. The sensitivity results of co-flow in Fig. 3.14(b)-(d) further show that it identifies more non-separate cases in the dynamic and static scenarios than in the hybrid scenario, achieving 85.06% sensitivity and 86.81% accuracy in the dynamic scenario and 87.50% sensitivity and 88.19% accuracy in the static scenario, compared to only 71.64% sensitivity and 81.25% accuracy in the hybrid scenario. Across all scenarios, the specificity of co-flow remains relatively stable, with 89.47% in the dynamic scenario, 88.89% in the static scenario, and 86.40% in the hybrid scenario.

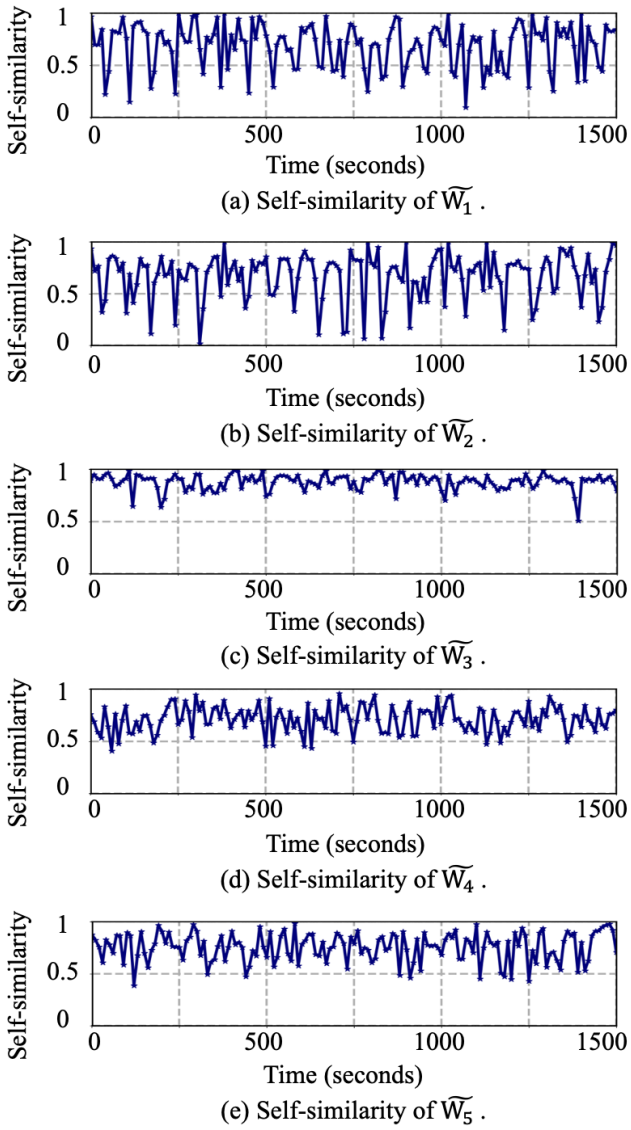
Based on the experimental results, we conclude that our metrics are more effective in detecting social distancing, particularly when devices rapidly change their proximity levels or carrying positions. Even when only Wi-Fi-based mobility similarity is used, our metrics remain sensitive to dynamic changes in network conditions. Therefore, the proposed metrics are well suited for applications that require real-time analysis.

Timeliness in Indicating Proximity Levels

We investigate whether cyber distances can effectively and promptly reflect relative proximity levels, spatial differences, and transitions between spaces. To this end, we first study the correlations between cyber distances and mobility similarities, including both self-similarities and inter-similarities, and then evaluate their correlations with movement similarities.

As shown in Fig. 3.15, the self-similarities of d_1 and d_2 fluctuate significantly due to device motion, whereas the self-similarities of d_3, d_4, and d_5 remain stable because they are static devices. This result demonstrates that the self-similarity metric can distinctly separate moving devices (d_1, d_2) from stationary devices (d_3, d_4, d_5). For the static devices, the self-similarities also capture differences in Wi-Fi network stability. As illustrated in Fig. 3.15(c)-(e), d_3 shows substantially higher stability than d_4 and d_5 because it is positioned in the right corridor, where more Wi-Fi APs are deployed and their RSS values are stronger than those in the left corridor. In conclusion, the self-similarity metric is capable of separating moving devices from static devices and further revealing the underlying stability of wireless network conditions that sensor data alone cannot capture.

Next, we investigate how the inter-similarity metric responds to changing spatial conditions to evaluate its ability to indicate proximity between devices. As shown in Fig. 3.16(a), during the first 900 seconds, the inter-similarity between d_1 and d_2 increases as they move closer together (approaching Position B) and decreases as they move farther apart. After 900 seconds, when d_2 is handed to the user carrying d_1, their inter-similarity remains consistently high because both devices follow the same trajectory (Trajectory 1). Similar results can be observed in Fig. 3.16(f) and Fig. 3.16(g), where the inter-similarity repeatedly rises and falls during the first 900 seconds as d_2 moves toward and away from the static devices d_4 and d_5 along Trajectory 2, and then drops sharply after 900 seconds when d_2 switches to Trajectory 1 and stays far from them. Moreover, Fig. 3.16(b) and Fig. 3.16(e) show clear round-trip patterns as both d_1 and d_2 repeatedly move toward and away from d_3, whereas the patterns in Fig. 3.16(c) and (d) are less pronounced because d_1 never approaches d_4 or d_5, resulting in consistently low inter-similarities. Note that the inter-similarity ranges in the first 900 seconds of Fig. 3.16(f) and Fig. 3.16(g) are much narrower than those in Fig. 3.16(e). This case arises because d_3 is located in the right corridor, where more APs are deployed and the RSS values are generally stronger, whereas d_4 and d_5 are located in the left corridor with fewer APs and weaker signals. With richer AP coverage and stronger RSS in the right corridor, the inter-similarity metric presents larger ranges (higher peaks and lower troughs), which makes proximity changes easier to capture. In contrast, the limited AP availability and weaker signals in the left corridor compress the ranges, resulting in smaller inter-similarity variations. Finally, Fig. 3.16(h)-(j) show that the inter-similarity between d_4 and d_5 is substantially higher than that between d_3 and d_5, demonstrating that the metric can effectively and promptly capture relative proximity levels, even



(a) Self-similarity of $\widetilde{W_1}$.

(b) Self-similarity of $\widetilde{W_2}$.

(c) Self-similarity of $\widetilde{W_3}$.

(d) Self-similarity of $\widetilde{W_4}$.

(e) Self-similarity of $\widetilde{W_5}$.

Figure 3.15: Self-similarity of each device in the complex environment.

under dynamic spatial transitions. These diverse experimental results indicate that pronounced round-trip patterns in inter-similarities naturally arise when two devices repeatedly move closer together and then farther apart. This demonstrates that the inter-similarity metric is highly sensitive to proximity fluctuations and serves as a reliable indicator of whether two devices are in close physical proximity. Moreover, even when devices are symmetrically positioned across a corridor (d_3 and d_5), the inter-similarity metric can still accurately determine whether they are co-located within the same space.

The movement similarities are presented in Fig. 3.17. As shown in Fig. 3.17(a), the movement similarity between d_1 and d_2 during the first 900 seconds exhibits

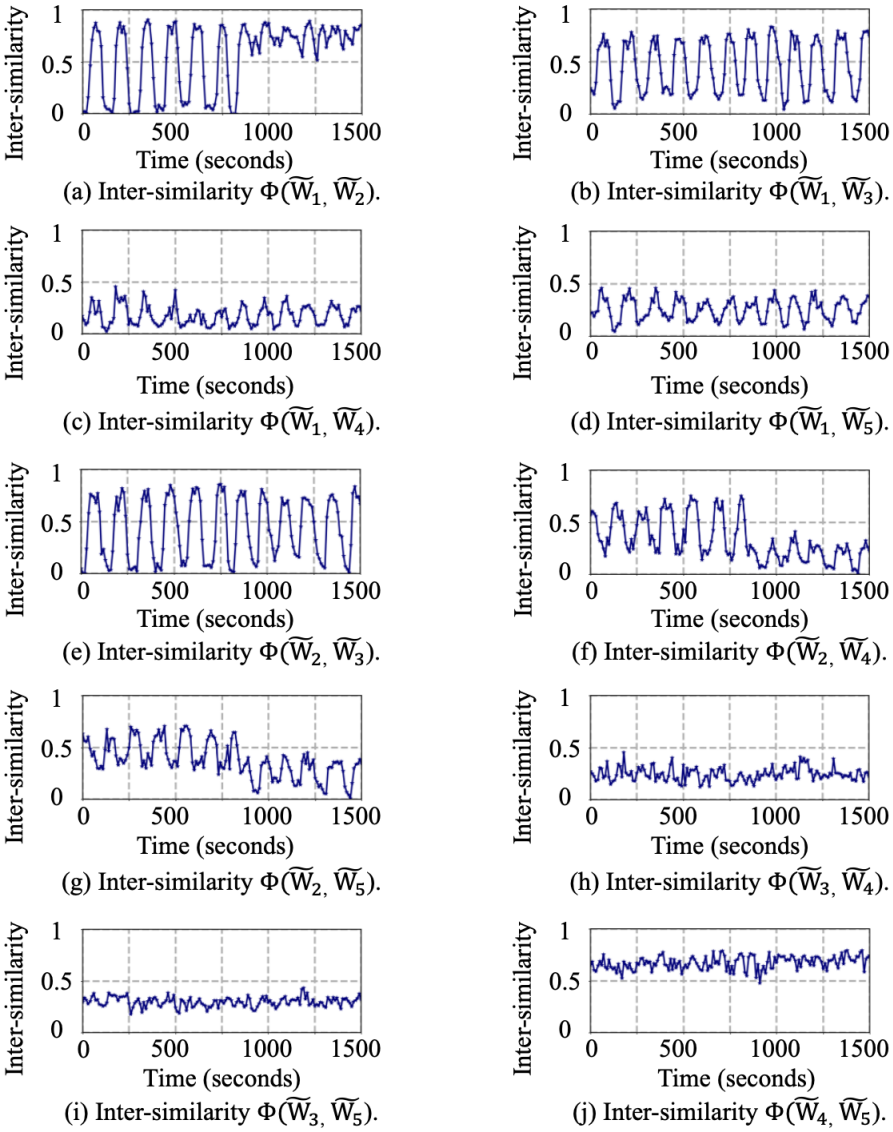


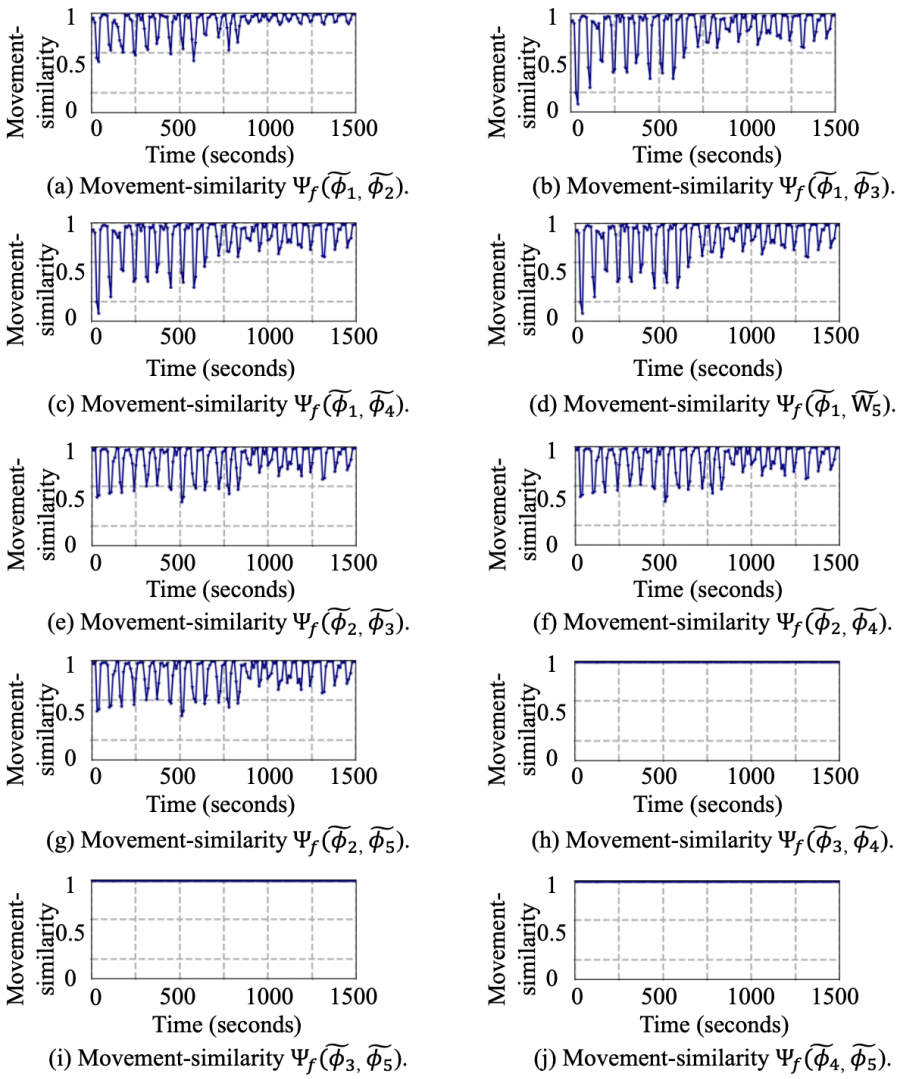
Figure 3.16: Inter-similarity between devices in the complex environment.

a sharp increase whenever both users arrive at the start or destination positions simultaneously and pause briefly to label the ground truth positions. Although their walking speeds are not perfectly synchronized, their movements along two equal-length trajectories remain loosely aligned. However, after 900 seconds, when d_2 is passed to another user, the movement similarity between d_1 and d_2 increases significantly, reflecting their close movement synchronization. Additionally, the variation in movement similarity narrows significantly. A similar trend is observed in Fig. 3.17(b), Fig. 3.17(c), and Fig. 3.17(d), where the range of movement similarity shrinks significantly after 900 seconds when d_1 is carried in a different way. Comparable results between d_2 and other stationary devices are depicted in Fig. 3.17(e),

Fig. 3.17(f), and Fig. 3.17(g). When all devices remain stationary, Fig. 3.17(h), Fig. 3.17(i), and Fig. 3.17(j) indicate very high movement similarities. Interestingly, even when d_1 and d_2 are held steadily in a user's palms to simulate a static state, the movement similarity metric can still differentiate whether the devices are truly stationary. These findings demonstrate that the movement similarity metric is highly sensitive to changes in device status, including variations in motion and how the devices are carried, even while in motion. This level of detail cannot be captured using Wi-Fi-based self-similarity and inter-similarity metrics. However, when all devices are stationary, the movement similarity metric alone cannot determine whether they share the same physical space. In such cases, Wi-Fi-based self-similarity and inter-similarity metrics become crucial for differentiation. In summary, the movement similarity metric effectively detects changes in device status, particularly when devices are in motion, regardless of how they are carried.

Combinations of Similarity Metrics and Parameters

In previous experiments, we investigate whether the proposed similarity metrics, i.e., inter-similarity, self-similarity, and movement similarity, can effectively and promptly capture relative proximity levels, spatial differences, and transitions between spaces. Accordingly, the ground truth requires only the overall movement patterns of the devices and does not record precise arrival and departure times at specific positions. In this section, we further evaluate how these metrics and their parameter selections influence system performance by measuring sensitivity, specificity, and accuracy. To achieve this, we study the correlations between the detection results of our system and the labeled arrival and departure timestamps at designated positions. For this purpose, we develop a mobile application to record arrival and departure times at predefined locations in the environment. As shown in Fig. 3.18(d), each user labels their current position by activating the corresponding button (i.e., "Position A," "Position B," or "Position C"). When departing from a position, the user deactivates the button to record the departure timestamp. When arriving at a new destination, the user activates the respective button to log the arrival timestamp. For example, when device d_1 arrives at Position A, the "Position A" button is activated to register the arrival time, and it is deactivated when d_1 leaves Position A. This procedure ensures accurate recording of all devices' arrival and departure timestamps, allowing us to determine the exact intervals during which a device is present at a given position. Fig. 3.18(a) and Fig. 3.18(b) show the labeled timestamps for d_1 and d_2 at Position A. Because the environment consists of a single straight corridor, deactivating all position buttons indicates that the device is traveling along the planned trajectory in the corridor. For a trajectory with two endpoints, if the labeled arrival and departure timestamps of two devices at these endpoints are well synchronized, the ground truth indicates that the devices are non-separate. In Fig. 3.18(a) and Fig. 3.18(b), after 900 seconds, the arrival and departure timestamps at Position A align perfectly for d_1 and d_2, indicating a non-separate ground truth.



(a) Movement-similarity $\Psi_f(\widetilde{\phi}_1, \widetilde{\phi}_2)$. (b) Movement-similarity $\Psi_f(\widetilde{\phi}_1, \widetilde{\phi}_3)$.

(c) Movement-similarity $\Psi_f(\widetilde{\phi}_1, \widetilde{\phi}_4)$. (d) Movement-similarity $\Psi_f(\widetilde{\phi}_1, \widetilde{W}_5)$.

(e) Movement-similarity $\Psi_f(\widetilde{\phi}_2, \widetilde{\phi}_3)$. (f) Movement-similarity $\Psi_f(\widetilde{\phi}_2, \widetilde{\phi}_4)$.

(g) Movement-similarity $\Psi_f(\widetilde{\phi}_2, \widetilde{\phi}_5)$. (h) Movement-similarity $\Psi_f(\widetilde{\phi}_3, \widetilde{\phi}_4)$.

(i) Movement-similarity $\Psi_f(\widetilde{\phi}_3, \widetilde{\phi}_5)$. (j) Movement-similarity $\Psi_f(\widetilde{\phi}_4, \widetilde{\phi}_5)$.

Figure 3.17: Movement similarity between devices in the complex environment.

The corresponding detection results in Fig. 3.18(c) are consistent with this ground truth, showing that d_1 and d_2 are detected as non-separate after 900 seconds.

We further evaluate the accuracy of our system by comparing the ground truth with the detection results under different combinations of similarity metrics. Fig. 3.19 presents the results as the time interval I increases from 5 to 15 and 25 seconds. When only a single metric is used, inter-similarity alone achieves the highest accuracy. When inter-similarity and self-similarity are combined, both the accuracy and its distribution are improved. Furthermore, incorporating movement similarity together with inter-similarity significantly enhances accuracy compared to using either Wi-Fi-based or sensor-based data alone. This improvement arises because Wi-Fi data and IMU data are complementary. First, movement similarity derived from IMU data

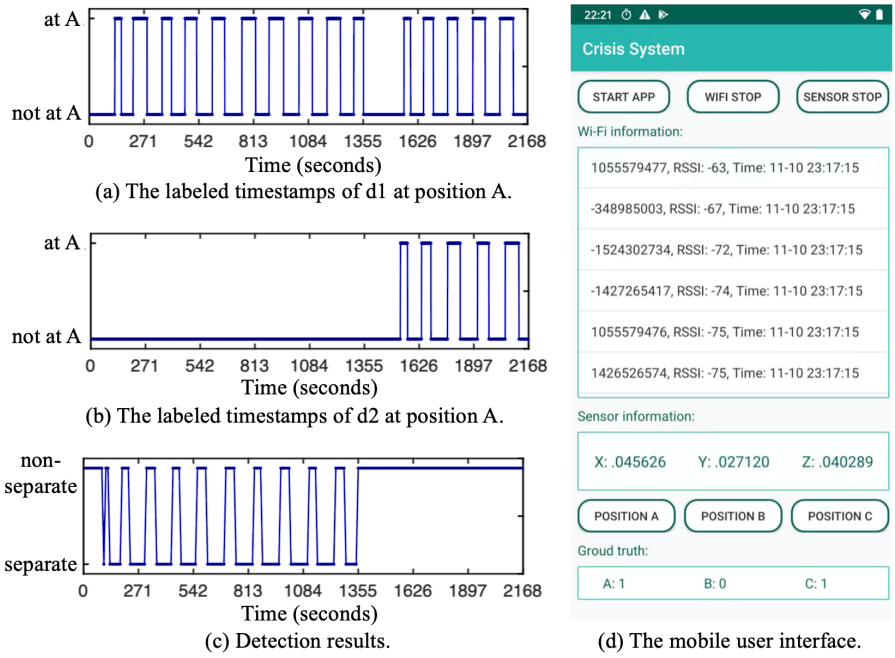


Figure 3.18: Validation with the correlated ground truth.

lacks explicit spatial information, and errors in location estimation may accumulate over time without absolute position references for calibration, ultimately degrading localization performance. Conversely, inter-similarity based on Wi-Fi data captures transitions between spaces and provides absolute spatial information. Second, Wi-Fi data can only provide coarse estimations of users' mobility, as it is less sensitive to small-scale movements or position changes. In comparison, IMU measurements can capture fine-grained motion dynamics and subtle movement variations with higher temporal resolution. When all similarity metrics (inter-similarity, self-similarity, and movement similarity) are combined and a longer time interval is used, the overall accuracy increases with the length of the time interval due to the availability of more sensing data. In addition, the accuracy distribution becomes more concentrated, and the average accuracy is slightly higher than that achieved by any other metric combination. This improvement is attributed to the contribution of self-similarity, which reflects both mobility stability and spatial characteristics, such as network conditions, that cannot be inferred from acceleration data alone, particularly when users remain stationary. This observation is further supported by the results shown in Fig. 3.15(c)-(e), as discussed in Section 3.6.5.

As defined in Section 3.6.2, the sensitivity, specificity, and accuracy of our system are computed from FP, TP, FN, and TN, which are binary results determined by predefined thresholds. The threshold for each similarity metric, i.e., inter-similarity, self-similarity, and movement similarity, is selected based on comprehensive experimental results. We use the results in Fig. 3.20 to explain the threshold selection process.

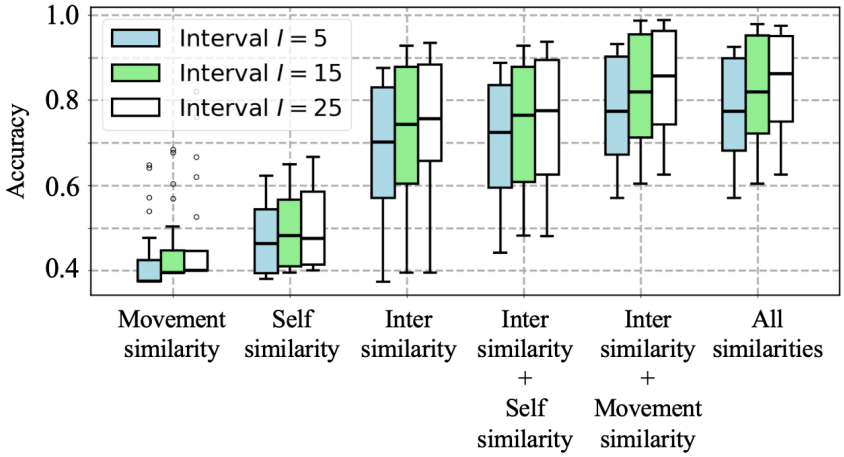
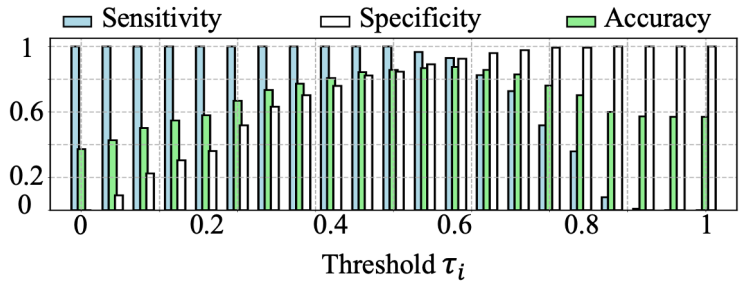


Figure 3.19: Accuracy of our system with different time intervals

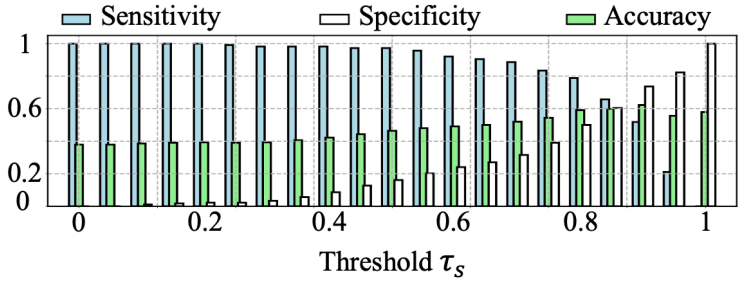
As shown in Fig. 3.20(a)-(c), clear trade-offs between sensitivity and specificity can be observed across all three figures, where a smaller threshold value yields higher sensitivity but lower specificity, while a larger threshold results in lower sensitivity but higher specificity. As explained in Section 3.6.2, sensitivity measures the ability to correctly identify non-separate users, whereas specificity measures the ability to correctly identify separate users. For applications where detecting non-separate users is more critical, such as monitoring whether users are maintaining social distancing, a smaller threshold is considered, while for applications where detecting separate users is more important, such as determining whether group members are still moving together, a larger threshold can be selected. Since the primary objective of this work is to detect non-separate users, a minimum sensitivity of 0.9 is enforced to ensure system robustness, and under this constraint the threshold is selected to maximize specificity. Accordingly, the thresholds are set to $\tau_i = 0.6$ for inter-similarity, $\tau_s = 0.5$ for self-similarity, and $\tau_f = 0.9$ for movement similarity, as shown in Fig. 3.20(a)-(c), respectively.

Energy and Communication Costs

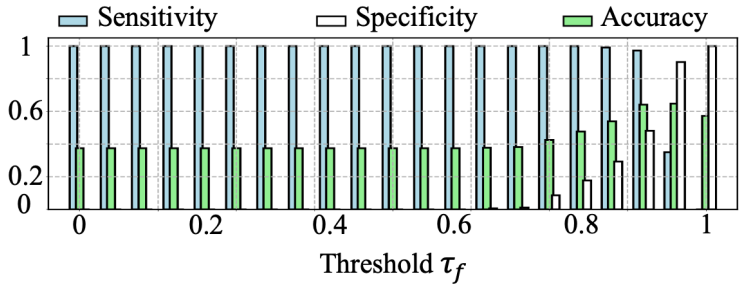
The energy consumption and communication cost of our system during runtime is further explored in this section. We consider device diversity with five different models and operating systems in the experiments. In the experiments, our mobile application runs continuously on these devices, collecting data iteratively over a period of 7.5 hours. The experimental setup follows the same configuration described in Section 3.5. All devices operate within the same room, where the number of detectable Wi-Fi APs typically remains below 25. The collected anonymized data is transmitted to a server via Wi-Fi networks. Table 3.2 presents the average energy consumption per hour and the corresponding communication cost per hour, as derived from Android Application Programming Interfaces (APIs). The



(a) Performance vs. τ_i selection for inter-similarity.



(b) Performance vs. τ_s selection for self-similarity.



(c) Performance vs. τ_f selection for movement similarity.

Figure 3.20: The impact of different thresholds on system sensitivity, specificity and accuracy.

estimated battery lifetime for each device is computed as the ratio of measured energy consumption to the full battery capacity of the respective device model. The results means that the maximum reported data size is 8.57 MB, while the highest recorded energy consumption is approximately 3.3% of the total battery capacity, equating to 97.16 mAh per hour. The variations in reported data sizes across devices stem from differences in hardware driver optimizations and operating system versions. Increasing the data collection interval I can further reduce both energy consumption and communication costs. Overall, our approach demonstrates energy efficiency in both sensing and data transmission, as only a minimal amount of data is required to be reported.

Table 3.2: Energy and communication costs for different devices.

	Data size for Communications	Energy Consumption	Operating System
Samsung Galaxy A5	7.82MB	69mAh *(2.38 % of lifetime)*	Android 7.0
Samsung Tab S4	7.91MB	170.3mAh *(2.33 % of lifetime)*	Android 8.0
Samsung Galaxy A8	8.57MB	58.6mAh *(1.95 % of lifetime)*	Android 9.0
Huawei P10 Plus	6.69MB	62.6mAh *(2.09 % of lifetime)*	Android 9.0
Google Pixel 3	8.38MB	97.16mAh *(3.3 % of lifetime)*	Android 11.0

3.6.6 Synchronicity between Cyber and Physical distances

The effectiveness of cyber distances in representing physical distances is evaluated based on two publicly available datasets, that include labeled coordinates collected from multiple devices in both 2D and 3D environments. Through extensive analyses, we examine the correlation between inter-similarities and physical distances across various combinations of 2D trajectories, different trajectory lengths, and 3D trajectories between multiple floors.

Dataset Descriptions and Statistics

Description of Dataset 1. This dataset [STJ+17] is collected on the first floor of a university building, covering an area of 185.12 m^2. The indoor environment consists of three rooms along a 40-meter-long corridor, with a total of 127 detectable Wi-Fi APs. This dataset contains Wi-Fi measurements taken at 324 distinct locations, each associated with labeled coordinates, and collected using two different devices for indoor localization, i.e., Sony Xperia M2 and LG W110G Watch R. Each device follows a predefined trajectory across the 324 locations, capturing a Wi-Fi sample at each point along the trajectory. Consequently, the dataset comprises a total of 628 Wi-Fi samples. Each sample records the Basic Service Set Identifiers (BSSID) and RSS values of the Wi-Fi APs detected at the corresponding labeled coordinate. These labeled coordinates serve as a reference for computing the physical distances between different trajectories in our analyses. For statistical evaluation, we randomly extract 300 sub-trajectories from each device's trajectory. Therefore, a total of 300×300 combinations of the two devices' sub-trajectories with different lengths (denoted by "2D mobility in Dataset 1") are obtained to conduct statistical analyses.

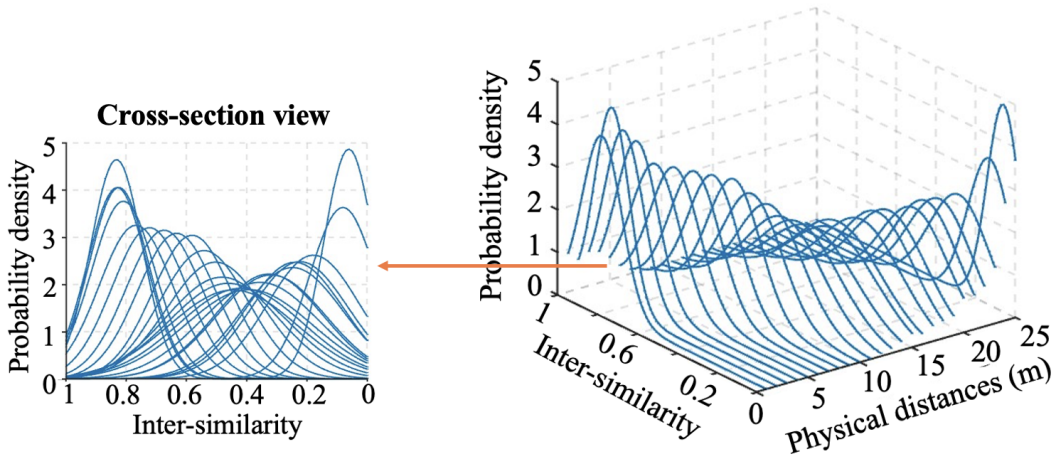
Description of Dataset 2. This dataset [BCL+16] is collected in a five-story university building comprising 822 irregular-shaped rooms. The total area of the

five floors is 22,570 m^2, where the height of each floor is 3.7 meters. This 3D dataset includes Wi-Fi measurements collected by 21 different devices at 3842 locations with labelled coordinates along their trajectories moving from one floor to the others. The 21 devices are heterogeneous in terms of types, brands, models, and operating systems. To analyze the impact of device heterogeneity, we generate different combinations of sub-trajectories using data collected both from the same device and from different devices. Specifically, we randomly select 300×300 pairs of 2D sub-trajectories,(denoted by "2D mobility in Dataset 2") and another 300×300 pairs of 3D sub-trajectories (denoted by "3D mobility in Dataset 2").

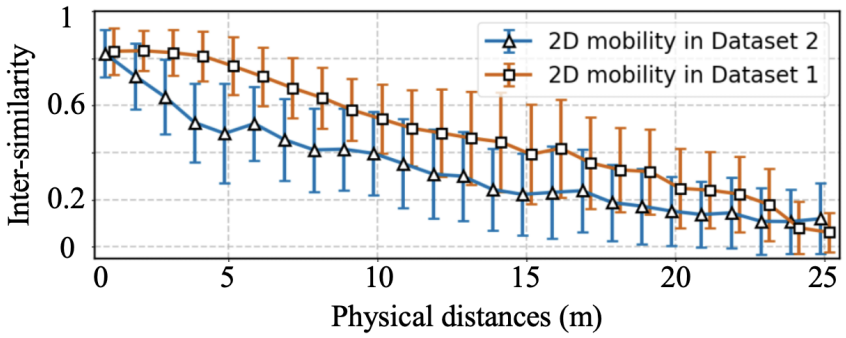
We compute the physical distances between sub-trajectories using the labelled coordinates. For each combination of two sub-trajectories, their first Wi-Fi samples are temporally aligned to compute their physical distance. However, since Wi-Fi samples are collected asynchronously, perfect alignment can not be guaranteed. To address this, the traditional DTW algorithm is applied to search for the shortest physical distance between them. Subsequently, for the 300×300 trajectory combinations in Dataset 1, we categorize them into 25 distinct groups based on their physical distances, ranging from $0 \sim 1$ m, $1 \sim 2$ m, ..., and $24 \sim 25$ m. Within each category, we compute the mean and standard deviation of the inter-similarity for the corresponding trajectory pairs. The same statistical procedure is also applied to the 2D and 3D sub-trajectory combinations derived from Dataset 2.

Extensive Analyses of 2D Trajectories

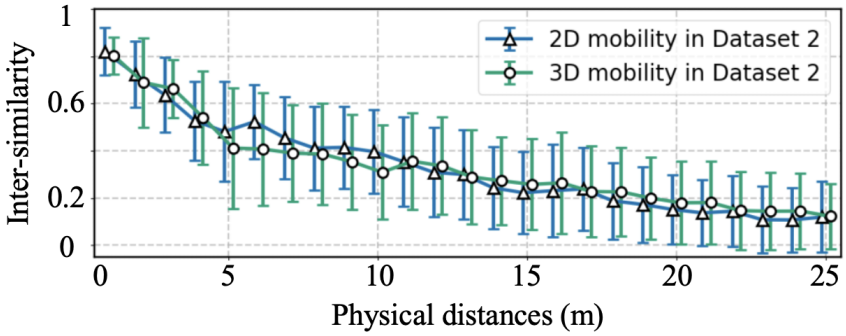
The correlation between distributions of inter-similarity and physical distances categories are presented in Fig. 3.21(a). Note that each physical distance category includes a range defined by its respective minimum and maximum values, which are determined in the previous statistical analysis. The results indicate that inter-similarity decreases as physical distance increases, demonstrating that the proposed inter-similarity metric effectively establishes a synchronous connection between cyber distances and physical distances. The standard deviations of the categories with very short (i.e., $0 \sim 5$ m) and very long ($20 \sim 25$ m) physical distances are relatively small. This is because, when two trajectories are either extremely close (or far apart) in the physical world, the sets of Wi-Fi APs detected along these trajectories exhibit significant similarities (or dissimilarities) in terms of the overlapping Wi-Fi APs and their RSS ranges. In this case, the generated mobility signatures demonstrate greater robustness in handling the absence of stronger Wi-Fi APs that have lower beacon broadcast rates, because most of the overlapping APs have similar weights. For physical distances ranging from 5 m to 20 m, greater variations in inter-similarity are observed. This is due to the vulnerability of RSS to environmental factors, such as signal attenuation caused by obstacles (e.g., walls). As a result, the overlapping Wi-Fi APs and RSS values between two trajectories exhibit significant variations. In these cases, missing stronger Wi-Fi APs with lower beacon broadcast rates results in significant fluctuations in inter-similarity.



(a) Statistics of 2D mobility in Dataset 1.



(b) Statistics of 2D mobility in two environments.



(c) Statistical analyses in a multi-floor environment.

Figure 3.21: Analyses of correlations between cyber distances and physical distances.

Furthermore, Fig. 3.21(b) presents the distributions of inter-similarities for 2D trajectories in two different environments, along with their respective means and standard deviations. The results indicate that inter-similarity decreases as the physical distance increases. The 2D mobility data from Dataset 2 exhibits lower inter-similarity due to the increased complexity of the environment, which contains many irregularly shaped rooms and obstacles (e.g., bookshelves arranged in rows and walls). In addition, the mobility combinations in Dataset 2 are generated from sub-trajectories collected by 21 heterogeneous devices, which increases data uncertainty. Note that, a sharp decline can be observed in inter-similarity as the physical distance increases from 1 m to 5 m. This is because many sub-trajectory pairs are spatially close but separated by walls, leading to substantial signal fading. This observation highlights that our metric captures not only the spatial distances between trajectories but also the physical-world space separation. In contrast, the inter-similarity curve for Dataset 1 remains relatively higher and more stable. This is because the sub-trajectories are made from the same device and the environment consists of three regular-shaped rooms, making it simpler in structure.

Extensive Analyses in a 3D Environment

The impacts of moving across floor in a 3D environment are further investigated here. The statistics of 2D mobility made on the same floor and statistics of 3D mobility taken between two adjacent floors are presented in Fig. 3.21(c). As can be seen, the differences between 2D and 3D curves are very small, which means that our approach is still robust even in a 3D environment. Note that the inter-similarity of 3D mobility is slightly lower within a physical distance between 4 m and 10 m, because the combination of the two sub-trajectories can potentially occur on two adjacent floors at a height of 3.7 m. In this case the ceiling greatly affects overlapping Wi-Fi APs and RSS in their mobility signatures because of signal fading. Our metric is also sensitive to space separation by different floors. When the physical distance increases to more than 10 m, Wi-Fi APs in their mobility signatures are almost not overlapping with each other because of the complex building structure and longer distances although they are on two adjacent floors.

3.6.7 Analyses of Execution Time

The theoretical and the actual execution time of our system is further investigated by varying the duration of time interval I. The communication cost during the data collection stage is not considered in the actual execution time, because only the computational cost of the algorithm is explored here. A longer I results in the larger numbers of Wi-Fi scans M and sensor measurements J, i.e., a larger M and J in the total complexity of $O(M^2N^2 + J \log J)$ analyzed in Section 3.4.2. In addition, the maximal number N of detectable Wi-Fi APs in an environment usually ranges from $5 \sim 10$ with sparse networks and ranges from $15 \sim 20$ with dense networks, as shown in Section 3.6.1 and Section 3.6.5. To analyze the impact of the value of N

on the actual execution time, the top $\tilde{N}$ Wi-Fi APs are selected based on RSS to create the mobility signature. Here, $\tilde{N}$ varies from 5, 10 to 15 to distinguish it from the actual N. To obtain the theoretical execution time, the theoretical values of M and J are approximated in Section 3.5 using $T_w = 1$ and $T_a = 1/50$ given I. Note that in practice, the values of M and J (depending on the optimization mechanisms implemented on different phone models) are much smaller than the approximated theoretical values.

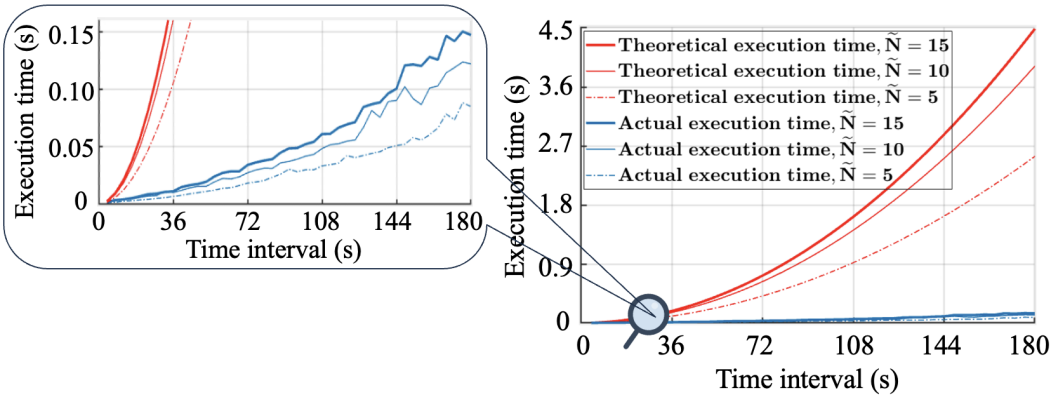


Figure 3.22: Time interval I vs. execution time.

The approximated theoretical execution time and the actual execution time are shown in Fig. 3.22. As expected, both theoretical and actual execution times in denser networks ($\tilde{N} = 15$) are significantly higher than those in sparser networks ($\tilde{N} = 10$ and $\tilde{N} = 5$). Furthermore, compared with the theoretical execution time (red curves), the actual execution time (blue curves) grows slowly with increasing time interval and remains significantly lower. This is because device manufacturers optimize the update cycles of Wi-Fi scanning and sensor sampling across operating system versions and device conditions, resulting in small values of M and J in practice. As shown in Fig. 3.22, when a short interval of $I = 6$ seconds is selected, which is the setting used in our implementation, the theoretical execution time is only 0.006 seconds even in dense networks with $\tilde{N} = 15$. In contrast, when a longer interval of $I = 180$ seconds is used, the theoretical execution time increases to 4.45 seconds in the same dense setting, consisting of 3.78 seconds for inter-similarity, 0.62 seconds for self-similarity, and 0.05 seconds for movement similarity. In practice, the actual execution time is much lower. In dense networks with $\tilde{N} = 15$, our system requires only 0.147 seconds for $I = 180$ seconds and merely 0.002 seconds for the implemented $I = 6$ seconds. Therefore, in our implementation, computing the cyber distance between two users requires only 0.002 seconds. Even when the number of users increases to 50, the total execution time remains small, i.e., $T = \binom{50}{2} \times 0.002 = \frac{50 \times 49}{2} \cdot 0.002 = 2.45$ seconds, which is much smaller the $I = 6$ seconds collection interval. Note that the execution time is tested on a server running on a Macbook Pro with a 2.3 GHz Intel i5 core processor. A more powerful machine,

such as a dedicated server, would further increase the number of users the system can support.

3.7 Conclusion

This dissertation designs a system to detect the physical proximity between users by analyzing their collected multi-modal sensing data. In this dissertation, the proposed system utilizes multi-modal sensing data (i.e., Wi-Fi data and IMU data without localization) to establish the mobility trajectories of users. Because Wi-Fi data serves as virtual landmark and IMU data can well perceive the user's motion, the mobility trajectories modeled for users in this dissertation also imply their real-world proximity to some extent. When the cyber distance between the mobility trajectories of any two users is computed to be small, the proposed system considers that these two users are also physically non-separate, i.e., they are not social distancing. The aim of our system is to utilize users' sensing data in the cyber world to infer their mobility in the physical world, thereby bridging the gap between the cyber and physical worlds. Extensive experimental results and in-depth statistical analysis based on comprehensive trajectories demonstrate that the proposed cyber distance is robust to the dynamic changes in physical distance caused by greater human mobility and movement. In addition, the joint use of multi-modal sensing data complements the weaknesses of Wi-Fi data and IMU data, respectively, providing a more reliable proximity detection method.

Chapter 4

Visual Trajectory Comparison Based on Cross-Domain Sensing Data

4.1 System Model

4.1.1 Problem Statement

This work analyzes the correlation between users' visual attention based cross-domain sensing data, i.e., the movements of human eyeballs and the light reflected in human eyes. First, we assume that each user is equipped with a camera for capturing consecutive images (video) containing the eyes of the user u_i. We divide time into multiple discrete intervals, and the length of each interval is M seconds. For a given interval, the video collected by the u_i over M seconds is denoted by $V_i = \{v_i(p)|p \in [1, MN]\}$, where $v_i(p)$ is the $p-th$ RGB frame containing the eye of u_i, and N is the number of frames per second (fps).

Given the videos V_i and V_j collected by two users u_i and u_j, respectively, this work designs a system to explore the visual attention correlation between u_i and u_j, referred to as the visual attention similarity $\Phi_{i,j}$ between u_i and u_j. As an example, a higher $\Phi_{i,j}$ means that the collected videos V_i and V_j are similar, also indicating that users u_i and u_j are always sharing a similar visual attention. Conversely, if u_i always focuses on one position while u_j is constantly changing his attention, their collected videos V_i and V_j are completely different, resulting in a smaller $\Phi_{i,j}$.

4.1.2 System Framework

Fig. 4.1 presents the framework of our system, which consists of four main phases: 1) data collection, 2) creation of visual signatures, 3) detection of visual transitions, and 4) computation of visual attention similarity.

In Phase 1, the image frames containing the eyes of a user are captured by his eye camera in real-time. The video consisting of sequential frames of this user is sent to the next phase for further processing. In Phase 2, iris of each user are extracted from each image frame in the video collected in Phase 1. Next, we create a visual signature for each processed frame by transforming each extracted iris from the spatial domain to the frequency domain. In Phase 3, this system detects the points where the visual signatures change. A change in the visual signatures implies that

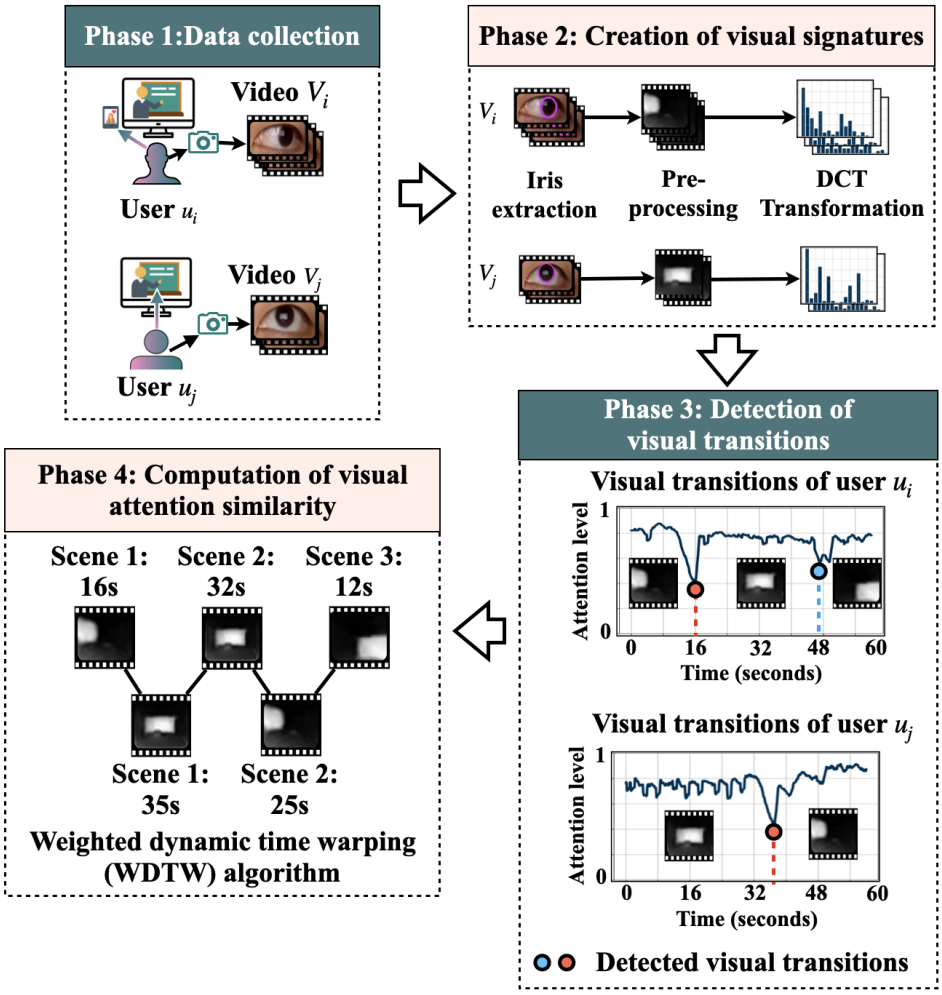


Figure 4.1: System framework.

the visual attention of the user changes from one position to another. We refer to these change points as the visual transitions, and the visual attention of the user is on different scenes before and after a visual transition. In Phase 4, based on the detected scenes, the visual attention similarity between users is computed through pairwise comparisons of the scenes seen by users. The more scenes two users simultaneously focus on and the closer the time they focus on each scene, the higher the visual attention similarity between the two users share.

4.2 Creation of Visual Signatures

This work creates a visual signature for each image frame by transforming each frame from the spatial domain to the frequency domain. The key idea behind the proposed visual signature lies in two aspects. First, the information in the

frequency domain can better characterize the light distribution reflected in users' eyes, where the high-frequency components represent the rapid changes of light whereas the low-frequency components stand for the slow changes of light. Second, salient features in low-resolution images, i.e., the light reflected in users' eyes, can be efficiently extracted from the frequency domain information, thus facilitating the effective comparison of low-resolution images.

This section first explains the technical details of constructing a visual signature, and then elaborates on how to quantify the similarity between visual signatures.

4.2.1 Workflow of Creating Visual Signatures

Given V_i, which is the video collected by u_i over M seconds, our system creates a visual signature for each image frame in V_i. The creation of the visual signatures for u_i is composed of 3 steps, which are presented in Fig. 4.2 and introduced as follows.

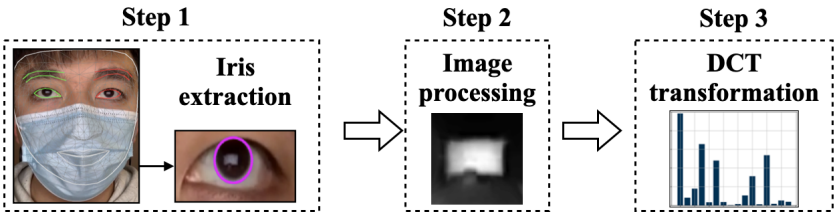


Figure 4.2: Creation of visual signatures.

In Step 1, the iris of u_i is extracted from each frame in V_i using the MediaPipe framework proposed in [GAK+20], which is based on a deep regression architecture. This architecture consists of a global stage and a local stage. In the global stage, a regression subnetwork estimates the coarse locations of facial landmarks, such as the mouth, nose, and iris, directly from the full face of u_i. In the local stage, each detected landmark is used as input to separate regression subnetworks to further refine the detection results. Since the detection output provides only the coordinates of the iris center, while our objective is to extract the iris region (i.e., the black part of the eye rather than the sclera), an additional processing step is required. Specifically, we adopt the method proposed in [Wel91], which applies a linear programming algorithm to find the smallest enclosing circle of the iris. The optimization objective of this linear programming algorithm is to minimize the radius of the circle while ensuring that all given points lie inside or on the boundary of the circle, which is achieved by jointly optimizing the circle center and radius subject to distance constraints imposed by the input points. Finally, we denote the extracted iris set by $E_i = \{e_i(p)|p \in [1, MN]\}$, where $e_i(p)$ is the iris region of user u_i extracted from the $p-th$ frame $v_i(p)$.

In Step 2, each image frame in E_i is first converted from RGB color space to HSV color space, where only the V component in HSV is leveraged in this work. This is because we investigate the light distribution reflected in human eyes and

the V component represents the lightness of an image. Moreover, since the contrast between light and the white part of the eye is lower, the light reflected in the eye whites is hard to utilize and may cause interference to the darker iris. To avoid extracting the eye whites, we convert the circular iris into a square through radial and linear stretching, so that the edge of the circle matches the edge of the square. We denote the processed irises by $\widetilde{E}_i = \{\widetilde{e}_i(p)|p \in [1, MN]\}$, where $\widetilde{e}_i(p)$ is the iris of u_i extracted from $v_i(p)$ after image processing.

In Step 3, each image frame in $\widetilde{E}_i$ is first transformed from the spatial domain to the frequency domain using discrete cosine transform (DCT) [ANR74]. The obtained DCT coefficients for each image frame are then sorted by frequency, and the lowest L DCT coefficients for each image frame are kept as the visual signature of this image frame. We denote by $D_i = \{d_i(p)|p \in [1, MN]\}$ the visual signatures of the user u_i, where $d_i(p) = \{\omega_1, \omega_2, ..., \omega_L\}$ is the visual signature generated for $\widetilde{e}_i(p)$, ω_1 is the DC coefficient with a frequency of zero and describes the overall lightness of $\widetilde{e}_i(p)$, and ω_2 to ω_L are the AC coefficients describing the changes of the lightness in $\widetilde{e}_i(p)$ at different frequencies.

4.2.2 Similarity Score between Visual Signatures

Given two visual signatures $d_i(p)$ and $d_i(p')$ created for $\widetilde{e}_i(p)$ and $\widetilde{e}_i(p')$, respectively, this work assumes that the more similar the light distributions in $\widetilde{e}_i(p)$ and $\widetilde{e}_i(p')$ are, the more overlapping frequencies in $d_i(p)$ and $d_i(p')$ are. To support this assumption, several examples of the visual signatures are presented in Fig. 4.3, where the length of L is set to 16. Fig. 4.3(a) and Fig. 4.3(b) depict the visual signatures of the user when the user looks at the right side of the monitor and the middle of the monitor, respectively. Fig. 4.3(c) and Fig. 4.3(d) both present the visual signatures of the user when the user looks at the left side of the monitor. As can be seen, the visual signatures in Fig. 4.3(a) and Fig. 4.3(b) exhibit completely different patterns because the user looks at different parts of the monitor. Conversely, the visual signatures Fig. 4.3(c) and Fig. 4.3(d) present similar patterns since the user looks at the same part of the monitor.

Based on the above assumption and observation, this work defines the similarity score between $d_i(p)$ and $d_i(p')$ in Eq. (4.1).

$$\Upsilon(d_i(p), d_i(p')) = \frac{\min(\Gamma(1, d_i(p)), \Gamma(1, d_i(p')))}{\max(\Gamma(1, d_i(p)), \Gamma(1, d_i(p')))} \times \frac{\sum_{l=2}^{L} \min(\Gamma(l, d_i(p)), \Gamma(l, d_i(p')))}{\sum_{l=2}^{L} \max(\Gamma(l, d_i(p)), \Gamma(l, d_i(p')))}; \tag{4.1}$$

where $\Gamma(l, d_i(p))$ is a function for extracting the *l-th* element from $d_i(p)$, as shown in Eq. (4.2).

$$\Gamma(l, d_i(p)) = \omega_l, \omega_l \in d_i(p); \tag{4.2}$$

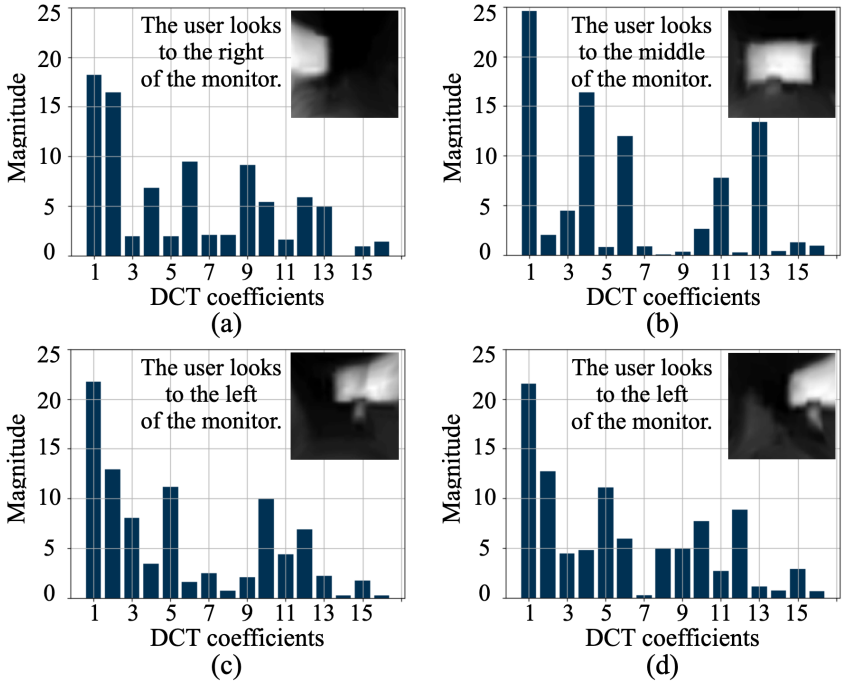


Figure 4.3: Visual signatures of u_i when u_i views different positions of the monitor.

The essence of Eq. (4.1) is to investigate the overlap level of $d_i(p)$ and $d_i(p')$. The first term in Eq. (4.1) investigates the overlap level of the DC coefficient between $d_i(p)$ and $d_i(p')$, while the second term in Eq. (4.1) investigates the overlap level of the AC coefficients between $d_i(p)$ and $d_i(p')$.

A numerical and graphical example of computing the similarity score is given in Fig. 4.4, where the similarity score between the visual signatures in Fig. 4.3(c) and Fig. 4.3(d) are computed. As shown in Fig. 4.4, the magnitudes of the DC coefficient is 21.7 in $d_i(p)$ and 21.5 in $d_i(p')$, respectively. The overlap of the DC coefficient is 21.5, which is computed by finding the minimum $\min(21.7, 21.5)$. The maximum $\max(21.7, 21.5)$ is the normalization factor controlling the overlap value between 0 and 1. Similarly, the overlap of the AC coefficients between $d_i(p)$ and $d_i(p')$ are computed. We first compute the overlap of each AC coefficient between $d_i(p)$ and $d_i(p')$, e.g., the overlap of the second coefficient is $\min(12.9, 12.8)$, the overlap of the third coefficient is $\min(8.1, 4.5)$, and the overlap of the last coefficient is $\min(0.3, 0.7)$. Next, these overlaps are summed up and referred to as the overlap of the AC coefficients between $d_i(p)$ and $d_i(p')$. Then, the overlap of the AC coefficients is normalized by a normalization factor, as shown in Eq. (4.1). Finally, the product of the overlap of the DC coefficients and the overlap of the AC coefficients is the similarity score between $d_i(p)$ and $d_i(p')$, which is 0.99×0.67=0.66 in this example. As a comparison, the similarity score between the visual signatures in Fig. 4.3(a) and Fig. 4.3(b) is 0.32, which is significantly smaller than 0.66.

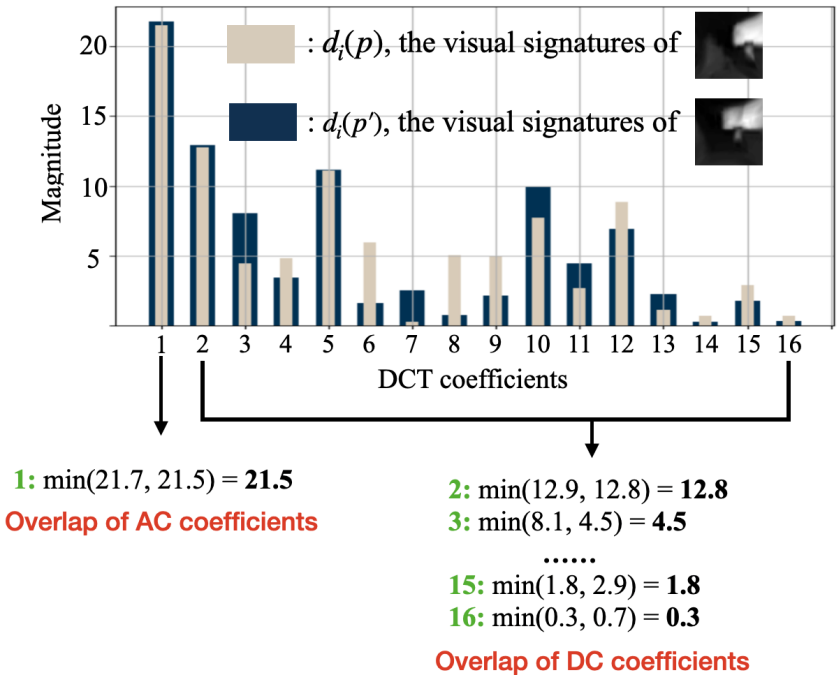


Figure 4.4: Illustration of similarity score between visual signatures.

4.3 Detection of Visual Transitions

This work first creates sequential visual signatures for each user, which implicitly represent their visual attention traces. Next, this work investigates the visual attention correlation between users by computing the similarity between their visual signatures. As the users' visual signatures are not synchronized, computing pairwise similarities between every two visual signatures not only provides coarse-grained analysis, but also imposes heavy computing burdens. Since the attention of a user is usually segmental in time, i.e., when his visual attention is focused on a certain place for a period of time, all his visual signatures during this period tend to be highly repetitive. As a result, this work segmentizes each user's visual attention into different scenes, by detecting the points where visual signatures change significantly (also referred to as visual transition points). Each detected scene is represented by a few visual signatures only. By comparing the scene-to-scene correlation, this work effectively investigates the visual attention correlation among users.

This section is organized into two parts. In the first part, this work quantifies the change level of the visual signatures of the user u_i within a certain time, where the change level of u_i is also referred to as the attention level of u_i. In the second part, the most significant changes are detected from the computed attention level of u_i, which serves as the visual transitions of u_i.

4.3.1 Computation of Attention Level

Rather than analyzing users' visual attention at a single time point, our system computes the attention level of each user by investigating temporally consecutive frames. This is because noise is unavoidable when capturing light reflected from the human eye. First, the iris of a user may be partially or completely occluded by the eyelid during eye blinking, resulting in incomplete or noisy visual information. Second, even when a user's attention remains focused on the same scene for an extended period, brief and involuntary eye movements, such as sudden glances, may occur and lead to erroneous visual signatures. To remain robust to such short-term noise and preserve only the most representative visual attention characteristics, the attention level of each user is computed by compensating individual erroneous visual signatures with other visual signatures collected within the same period.

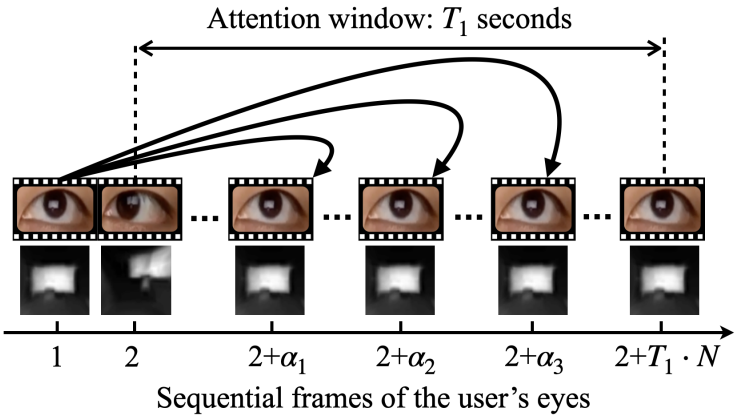


Figure 4.5: Illustration of attention window.

Given $D_i = \{d_i(p)|p \in [1, MN]\}$, which represents the visual signatures of user u_i, the attention level of u_i is computed by quantifying changes in visual signatures over an attention window. As shown in Fig. 4.5, $d_i(p)$ denotes the visual signature collected at a given time, and the subsequent T_1 seconds are defined as the attention window associated with $d_i(p)$. First, a set $\Psi_i = \{\psi_i(p')|p' \in [1, (M - T_1)N]\}$ is initialized to store the attention levels corresponding to the visual signatures of u_i, where $\psi_i(p')$ denotes the attention level of $d_i(p')$ over its attention window. To compute $\psi_i(p')$, we randomly select K visual signatures $\{\alpha_1, ..., \alpha_K\}$ from the attention window of $d_i(p')$. Next, $\psi_i(p')$ is updated by computing the average similarity score between these K visual signatures and $d_i(p')$, denoted by $\psi_i(p') = \frac{\sum_{k=1}^{K} \Upsilon(d_i(p'), d_i(\alpha_k))}{K}$. Finally, a moving average filter with window length T_2 is applied to the resulting attention sequence to further suppress noise.

4.3.2 Detection of Visual Transitions

As stated in Section 4.3, the attention level changes significantly when a user's visual attention shifts from one scene to another, whereas it remains relatively stable when the user's visual attention stays on the same scene. Therefore, a user's visual attention can be segmented into different scenes by identifying significant changes in the attention level. Given the attention-level sequence Ψ_i of user u_i, this section explains how to detect the most significant changes in Ψ_i, which are referred to as visual transitions and denoted as $T_i = \{t_i(1), t_i(2), \ldots, t_i(l), \ldots\}$.

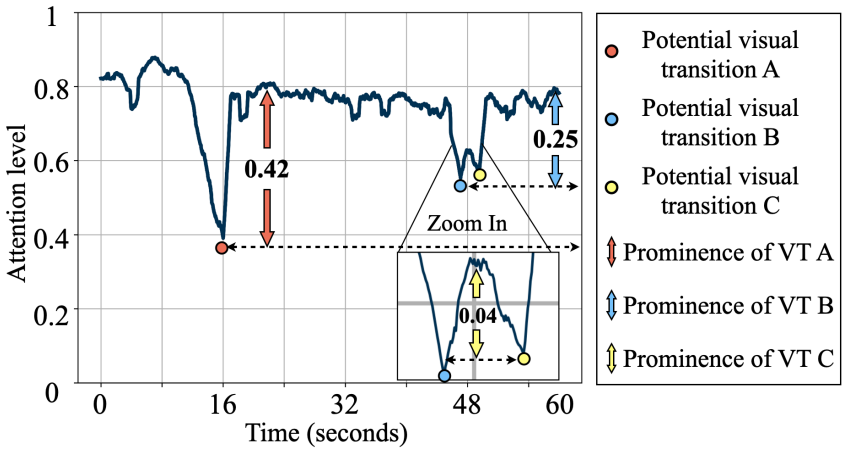


Figure 4.6: Detection of visual transitions for user u_i.

In this work, visual transitions are detected based on three criteria derived from empirical observations, which are explained using the examples shown in Fig. 4.6. First, the attention level must be smaller than a predefined threshold τ_1, which is set to 0.6 in our system. Second, the attention level must correspond to a local minimum. As shown in Fig. 4.6, only three candidate points satisfy these two criteria and are preserved as potential visual transitions, denoted as points A, B, and C, and marked by red, yellow, and blue dots, respectively. The third criterion evaluates whether the prominences of these candidate points exceed τ_2. We explain how to find the prominence of a point based on the examples in Fig. 4.6. First, a horizontal dashed line is extended from point A to the right until it intersects higher attention levels or reaches the boundary. The maximum vertical distance between the attention level and this dashed line is denoted as $r_{\max}$. Similarly, a dashed line is extended from point A to the left, and the maximum vertical distance between the attention level and this dashed line is denoted as $l_{\max}$. The prominence of point A is defined as the minimum of $r_{\max}$ and $l_{\max}$, which is 0.42 in Fig. 4.6. Based on the same procedure, the prominences of points B and C are computed as 0.25 and 0.04, respectively. Since τ_2 is empirically set to 0.2, point C does not satisfy the third criterion and is therefore not identified as a visual transition. As a result, points A and B are selected as actual visual transitions.

The key idea of prominence in visual transition detection is to eliminate scenes that attract only brief user attention while preserving scenes that represent the user's actual focus. Specifically, when a user's visual attention switches rapidly between scenes, multiple candidate visual transitions may be detected within a short time interval, as shown by points B and C in Fig. 4.6. The prominence criterion identifies the most representative visual transition by assessing the correlations between potential visual transitions, suppressing transient transitions (position C) and preserving only the actual visual transition (Position B). In contrast, point A corresponds to a distinctive visual transition, exhibiting sufficiently high prominence and thus being correctly retained as an independent scene change.

4.4 Quantification of Visual Attention Similarity

Based on the visual transition T_i detected in Section 4.3, the visual attention of u_i is divided into multiple segments. We denote these segments by S_i, which is a time series containing the scenes seen by u_i over a period of time. Given S_i and S_j for users u_i and u_j, this section analyzes the visual attention similarity $\Phi_{i,j}$ between u_i and u_j by pairwise comparing the scenes between S_i and S_j. The more similar scenes are in S_i and S_j, the higher the visual attention similarity between u_i and u_j is. The comparisons between S_i and S_j are made by an algorithm called Weighted Dynamic Time Warping (WDTW) designed in this work. The key idea of the WDTW algorithm lies in two aspects. First, the WDTW algorithm searches for the optimal matching between S_i and S_j by maximizing their similarity score defined in Eq. (4.1), while the traditional DTW algorithm finds the best matching between time series by minimizing their Euclidean distance. Second, each scene in S_i and S_j are assigned different weights in the WDTW algorithm, whereas all elements are of equal importance in the traditional DTW algorithm.

The proposed WDTW algorithm consists of two parts. The first part is shown in Algorithm 2, where the pairwise similarity scores of the scenes between S_i and S_j are first computed. Then, the computed similarity scores are weighted based on the duration of each scene. The second part is shown in Algorithm 3, where the WDTW algorithm first searches for the optimal path with the largest accumulation of the weighted similarity scores, and then retrieves the optimal matching from the optimal path. The visual attention similarity $\Phi_{i,j}$ is computed based on the optimal matching.

4.4.1 Computation of Pairwise Similarity Scores

Algorithm 2 pairwise compares the scenes between users while considering the duration of each scene. As shown in Lines 2-10 in Algorithm 2, the scenes of u_i and u_j are denoted by S_i=$\{s_i(p)|p \in (1, |T_i| - 1)\}$ and S_j=$\{s_j(q)|p \in (1, |T_j| - 1)\}$, where $s_i(p)$ is the *p-th* scene of u_i and $s_j(q)$ is the *q-th* scene of u_j. The comparison between $s_i(p)$ and $s_j(q)$ is first made by computing the similarity score of the

visual signatures between $s_i(p)$ and $s_j(q)$, as shown in Lines 11-16. The computed similarity score is kept in the matrix $G_{i,j}$. Then, $G_{i,j}$ is weighted by importance level δ_1 and equality level δ_2, as shown in Lines 17-19. The importance level of $s_i(p)$ and $s_j(q)$ is defined as $\frac{|s_i(p)|+|s_j(q)|}{2MN}$, i.e., the ratio of the durations of $s_i(p)$ and $s_j(q)$ to the total duration. The equality level of $s_i(p)$ and $s_j(q)$ is defined as $\frac{min(|s_i(p)|,|s_j(q)|)}{max(|s_i(p)|,|s_j(q)|)}$, i.e., the ratio of the shorter-duration scene to the longer-duration scene in this pair of scenes. The weighted similarity scores are denoted by the matrix $\widetilde{G}_{i,j}$.

Fig. 4.7 presents a numerical example to better illustrate the principals of Algorithm 2. As shown in Fig. 4.7(a), three scenes $\{s_i(1), s_i(2), s_i(3)\}$ are detected for u_i and two scenes $\{s_j(1), s_j(2)\}$ are detected for u_j. Note that, each scene is represented as the range between two visual transitions, e.g., $s_i(1)$ is the range between the visual transitions $t_i(1)$ and $t_i(2)$. Then, pairwise comparisons between $\{s_i(1), s_i(2), s_i(3)\}$ and $\{s_j(1), s_j(2)\}$ are performed. An example of comparing $s_i(2)$ and $s_j(1)$ is given. Firstly, K visual signatures are randomly selected from $s_j(1)$, denoted by $\{d_i(\alpha_1), ..., d_i(\alpha_K)\}$. Similarly, K visual signatures are also randomly selected from $s_i(2)$, denoted by $\{d_j(\beta_1), ..., d_j(\beta_K)\}$. Secondly, we compute the pairwise similarity scores of the visual signatures between $\{d_i(\alpha_1), ..., d_i(\alpha_K)\}$ and $\{d_j(\beta_1), ..., d_j(\beta_K)\}$, and average the computed similarity scores. The averaged similarity score of scenes $s_i(2)$ and $s_j(1)$ is denoted by $G_{i,j}(1,2)$, which is computed as 0.79. As shown in Fig. 4.7(b), $s_i(2)$ and $s_j(1)$ are the similar scenes, which is in accordance with the higher similarity score $G_{i,j}(1,2) = 0.79$. Thirdly, $G_{i,j}(1,2)$ is weighted by the importance level δ_1 and the equality level δ_2. As the durations of $s_i(2)$ and $s_j(1)$ are 32 seconds and 34 seconds, δ_1 is computed as $\frac{32+34}{60+60} = 0.55$ and δ_2 is computed as $\frac{32}{34} = 0.94$ for $s_i(2)$ and $s_j(1)$. Therefore, the weighted similarity score $\widetilde{G}_{i,j}(2,1)$ of $s_i(2)$ and $s_j(1)$ is $0.79 \times 0.55 \times 0.94 = 0.40$, as shown in Fig. 4.7(c).

The meanings behind the importance level and the equality level are exemplified by another example in Fig. 4.7. Because $s_i(1)$ and $s_j(2)$ are also the similar scenes, the averaged similarity score of $s_i(1)$ and $s_j(2)$ is $G_{i,j}(1,2) = 0.87$, which is higher than 0.79 for $G_{i,j}(2,1)$. However, as the durations of $s_i(1)$ and $s_j(2)$ are shorter, their importance level is merely $\frac{16+26}{120} = 0.35$. In addition, the durations $s_i(1)$ and $s_j(2)$ are different, their equality level is only $\frac{16}{26} = 0.62$. As a result, the weighted similarity score $\widetilde{G}_{i,j}(1,2)$ of $s_i(1)$ and $s_j(2)$ is $0.87 \times 0.35 \times 0.62 = 0.19$ which is significantly smaller than $\widetilde{G}_{i,j}(2,1)$. Therefore, a larger weight is assigned to a pair of scenes when the duration of the pair is longer and the durations of two scenes in the pair are closer.

Algorithm 2: Computation of Pairwise Similarity Scores

Input : T_i, T_j: Visual transitions of users u_i and u_j;
D_i, D_j: Visual signatures collected by users u_i and u_j;

Output : S_i, S_j: Scenes seen by users u_i and u_j;
$G_{i,j}$: Pairwise similarity scores of scenes between users u_i and u_j;
$\widetilde{G}_{i,j}$: Weighted pairwise similarity scores of scenes between users u_i and u_j;

//**Initialization:**
$S_i = \{s_i(1), ..., s_i(|T_i|-1)\} \leftarrow 0$;
$S_j = \{s_j(1), ..., s_j(|T_j|-1)\} \leftarrow 0$;
$G_{i,j} \leftarrow 0_{|T_i|-1,|T_j|-1}$;
$\widetilde{G}_{i,j} \leftarrow 0_{|T_i|-1,|T_j|-1}$;
//**Compute the range of each scene:**
for *p from 1 to* $|T_i|-1$ **do**
  $s_i(p) \leftarrow [t_i(p), t_i(p+1)]$;
for *q from 1 to* $|T_j|-1$ **do**
  $s_j(q) \leftarrow [t_j(q), t_j(q+1)]$;
//**Compute weighted similarity scores:**
for *p from 1 to* $|T_i|-1$ **do**
  $\{\alpha_1, ..., \alpha_K\} \leftarrow$ Randomly select K frames from $s_i(p)$;
  for *q from 1 to* $|T_j|-1$ **do**
    $\{\beta_1, ..., \beta_K\} \leftarrow$ Randomly select K frames from $s_j(q)$;
    $G_{i,j}(p,q) \leftarrow \frac{\sum_{u=1}^{K}\sum_{v=1}^{K}\Upsilon(d_i(\alpha_u), d_j(\beta_v))}{K \times K}$; //Compute the similarity score between $s_i(p)$ and $s_j(q)$.
    $\delta_1 \leftarrow \frac{|s_i(p)| + |s_j(q)|}{2MN}$; //Importance level.
    $\delta_2 \leftarrow \frac{min(|s_i(p)|, |s_j(q)|)}{max(|s_i(p)|, |s_j(q)|)}$; //Equality level.
    $\widetilde{G}_{i,j}(p,q) \leftarrow \delta_1 \times \delta_2 \times G_{i,j}(p,q)$; //Compute the weighted similarity score between $s_i(p)$ and $s_j(q)$.
return $G_{i,j}, \widetilde{G}_{i,j}, S_i, S_j$

4.4.2 Computation of Visual Attention Similarity

Given $\widetilde{G}_{i,j}$, Algorithm 3 searches for the optimal matching of the scenes between S_i and S_j such that the accumulation of the similarity scores is maximal. To achieve this, Algorithm 3 first computes the accumulation of the similarity scores for each element in $\widetilde{G}_{i,j}$. The accumulations of the similarity scores are kept in the matrix $P_{i,j}$. Then, the optimal matching is found by searching for the optimal path with the largest accumulation of the similarity score from $P_{i,j}$.

An example is given in Fig. 4.8 to demonstrate how Algorithm 3 works, where $P_{i,j}$ is indicated by the red numbers in Fig. 4.8(a), and $\widetilde{G}_{i,j}$ is indicated by the numbers in brackets in Fig. 4.8(a). Algorithm 3 starts by computing $P_{i,j}(1,1)$ for

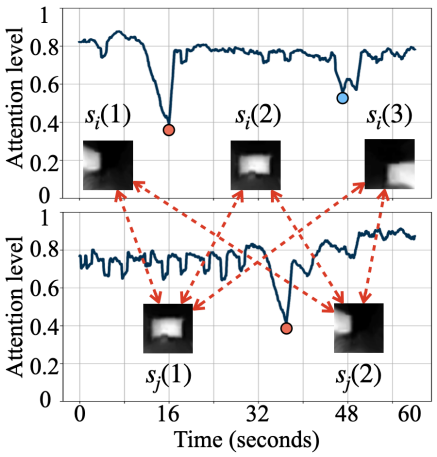
Algorithm 3: Computation of Visual Attention Similarity

Input : S_i, S_j: Scenes of users u_i and u_j;
$G_{i,j}$: Similarity scores between S_i and S_j;
$\widetilde{G}_{i,j}$: Weighted similarity scores between S_i and S_j;
Output : $\Phi_{i,j}$: Visual attention similarity between u_i and u_j;

//**Initialization:**
$P_{i,j} \leftarrow 0_{|T_i|-1,|T_j|-1}$; //Accumulation of similarity score.
$O \leftarrow \varnothing$; //Set for keeping optimal matching.
//**Compute the maximal accumulation of similarity score:**
$P_{i,j}(1,1) \leftarrow 2 \times \widetilde{G}_{i,j}(1,1)$; //Starting point.
for *p from 2 to* $|T_i|-1$ **do**
  $P_{i,j}(p,1) \leftarrow P_{i,j}(p-1,1) + \widetilde{G}_{i,j}(p,1)$; //First column.
for *q from 2 to* $|T_j|-1$ **do**
  $P_{i,j}(1,q) \leftarrow P_{i,j}(1,q-1) + \widetilde{G}_{i,j}(1,q)$; //First row.
for *p from 2 to* $|T_i|-1$ **do**
  for *q from 2 to* $|T_j|-1$ **do**
    $v\_path \leftarrow P_{i,j}(p-1,q) + \widetilde{G}_{i,j}(p,q)$;
    $h\_path \leftarrow P_{i,j}(p,q-1) + \widetilde{G}_{i,j}(p,q)$;
    $d\_path \leftarrow P_{i,j}(p-1,q-1) + 2 \times \widetilde{G}_{i,j}(p,q)$;
    $P_{i,j}(p,q) \leftarrow max(v\_path, h\_path, d\_path)$;
//**Backtrack to search for the optimal matching:**
while $p > 1$ *or* $q > 1$ **do**
  if $p = 1$ **then**
    $o \leftarrow (1, q-1)$
  else if $q = 1$ **then**
    $o \leftarrow (p-1, 1)$
  else
    if $P_{i,j}(p,q) = P_{i,j}(p-1,q) + \widetilde{G}_{i,j}(p,q)$ **then**
      $o \leftarrow (p-1, q)$
    else if $P_{i,j}(p,q) = P_{i,j}(p,q-1) + \widetilde{G}_{i,j}(p,q)$ **then**
      $o \leftarrow (p, q-1)$
    else
      $o \leftarrow (p-1, q-1)$
  $O \leftarrow O \cup \{o\}$
  $(p,q) \leftarrow o$
//**Compute the visual attention similarity :**

$$\Phi_{i,j} \leftarrow \frac{\sum_{p=1}^{|O|} G_{i,j}(x_1,y_1) \times (|s_i(x_1)| + |s_j(y_1)|), (x_1,y_1) \in O(p)}{\sum_{q=1}^{|O|} |s_i(x_2)| + |s_j(y_2)|, (x_2,y_2) \in O(q)}$$

return $\Phi_{i,j}$



(a) Pairwise comparisons of scenes between S_i and S_j.

	$s_i(1)$	$s_i(2)$	$s_i(3)$
$s_j(1)$	0.38	0.79	0.51
$s_j(2)$	0.87	0.44	0.35

(b) Similarity score of scenes between S_i and S_j.

	$s_i(1)$	$s_i(2)$	$s_i(3)$
$s_j(1)$	0.07	0.40	0.07
$s_j(2)$	0.19	0.16	0.05

(c) Weighted similarity score of scenes between S_i and S_j.

Figure 4.7: Illustration of attention window.

the first element in $\widetilde{G}_{i,j}$ according to Line 5, which is $2 \times 0.07 = 0.14$. Then, the accumulations of the similarity scores are computed for the elements in the first column and first row. For example, $P_{i,j}(2,1)$ is computed according to Lines 6-7, which is $0.14 + 0.40 = 0.54$. Similarly, $P_{i,j}(1,2)$ and $P_{i,j}(1,3)$ are computed according to Lines 8-9, which are 0.54 and 0.61. The similarity scores are accumulated along only one path for the elements in the first row or first column, whereas three possible paths are examined for each of the other elements in $\widetilde{G}_{i,j}$, as shown in Lines 10-15. For example, the three possible paths are examined for computing $P_{i,j}(2,2)$, which are the vertical path $P_{i,j}(1,2) \rightarrow \widetilde{G}_{i,j}(2,2)$, the horizontal path $P_{i,j}(2,1) \rightarrow \widetilde{G}_{i,j}(2,2)$, and the diagonal path $P_{i,j}(1,1) \rightarrow \widetilde{G}_{i,j}(2,2)$. The maximum among the three paths is selected for computing $P_{i,j}(2,2)$, which is the vertical path $0.54 + 0.16 = 0.70$. Similarly, $P_{i,j}(2,3)$ is computed as $0.70 + 0.05 = 0.75$. Next, Algorithm 3 starts from the last element $P_{i,j}(2,3)$ and backtracks to search for the optimal path from $P_{i,j}$, as shown in Lines 17-28. The optimal path is $0.14 \rightarrow 0.54 \rightarrow 0.70 \rightarrow 0.75$, as shown by the red arrows in Fig. 4.8(b). The optimal matching is retrieved from the optimal path, which are $s_i(1)$ and $s_j(1)$, $s_i(2)$ and $s_j(1)$, $s_i(2)$ and $s_j(2)$, $s_i(3)$ and $s_j(2)$, as shown in Fig. 4.8(b). Finally, the visual attention similarity $\Phi_{i,j}$ is found by normalizing the similarity scores between the scenes in the optimal matching, as shown in Line 32.

4.5 Implementation

Users can decide the operation mode of our system according to their application scenarios. In static scenarios such as online classes, our plug-and-play system is installed on the user's computer. The eyes of the user are captured by the webcam of the computer. In dynamic scenarios (e.g., some virtual reality-based applications),

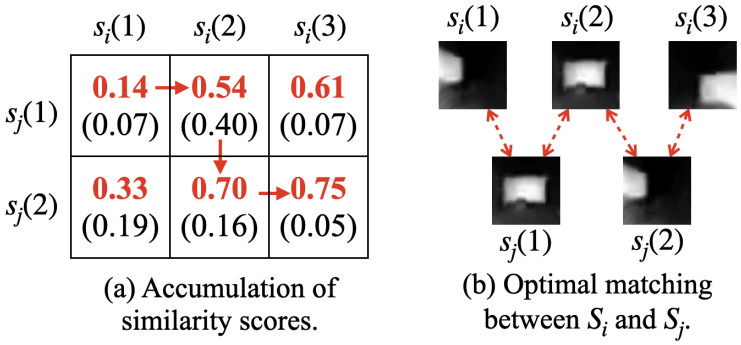


Figure 4.8: An example of optimal matching.

the eye camera mounted on the user's smart glasses can capture the light reflected in their eyes. As shown in Fig. 4.9, the prototype of the wearable device consists of four modules: i.e., an IMX477 camera for capturing the eyes of the user, an Nvidia Jetson Nano board for running our system, a TL-WN722N Wi-Fi antenna for data transmission, a helmet and 3D printed adjustable holder for fastening all modules.

Figure 4.9: Prototype of the designed wearable device.

The proposed parameters are specified in this section. The time interval M is set to 60 seconds, i.e., this work investigates the attention of users for a period of 60 seconds. The frames per second N is set to 30 for capturing the fast eye movements. The length of the preserved DCT coefficients L is set to 16, which is sufficient for the comparisons of the low-resolution images in this work. T_1 for the attention window and T_2 for the sliding window are set to 5 seconds for noise removal. The thresholds τ_1 and τ_2 for detecting the visual transitions are set to 0.6 and 0.2, respectively.

4.6 Experiments and Evaluation

This section evaluates the performance of the proposed system through extensive experiments conducted in three environments, i.e., staring at computer screens, indoors in university buildings, and outdoors on campus. We divide the experiments into controlled experiments and uncontrolled experiments. The controlled exper-

iments are conducted under different attention scenarios, which are designed by simulating different users' visual attention. By comparing the detected users' visual attention with their actual visual attention, we preliminarily investigate the ability of the proposed system. Moreover, this work provides an in-depth analysis of the system performance under diverse attention scenarios, different environments, and various external factors. The uncontrolled experiments are conducted on a large scale to further verify the robustness of the proposed system. To achieve this, we investigate the synchronicity between the visual attention similarity and the overlap ratio of users' attention. Finally, the necessity of some parameters proposed in this work is also elaborated through uncontrolled experiments.

4.6.1 Preliminary Analysis based on Controlled Experiments

Experimental Setup

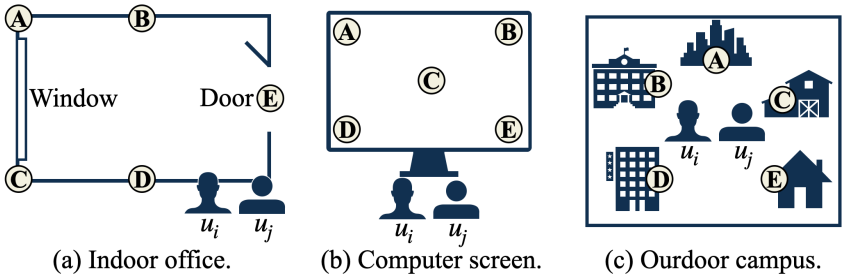


Figure 4.10: Experimental environments.

This section first conducts controlled experiments in different environments and under diverse attention scenarios. As shown in Fig. 4.10, The experimental environments are an indoor office, a computer screen, and the outdoor campus in this work. For each environment, 5 reference points are selected from different positions which are marked by A, B, C, D, E in Fig. 4.10. Users' attention only switches between reference points in controlled experiments. As shown in Fig. 4.11, 6 different attention scenarios are designed by simulating various visual attention for two users u_i and u_j. We conduct controlled experiments under 6 attention scenarios in each environment, the detail of each attention scenario is described as follows.

- **Scenario 1**: both u_i's attention and u_j's attention are on one position (e.g., Position A) for a period of 400 seconds, as shown in Fig. 4.11(a);
- **Scenario 2**: u_i's attention is on one position (e.g., Position A) while u_j's attention is on a different position (e.g., Position B) for 400 seconds, as shown in Fig. 4.11(b);
- **Scenario 3**: both u_i and u_j change their attention from one position to another every 10 seconds, their attention is always on the same position for 400 seconds, as shown in Fig. 4.11(c);

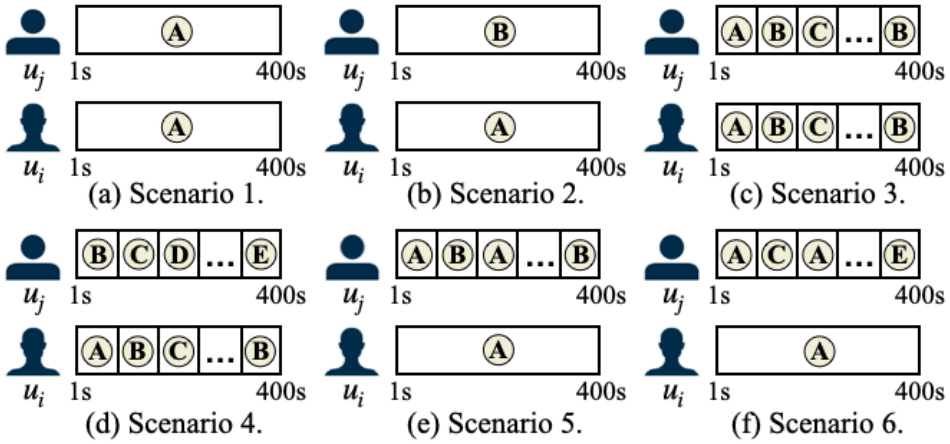


Figure 4.11: Experimental scenarios.

- **Scenario 4**: both u_i and u_j change their attention from one position to another every 10 seconds, their attention is always on different positions for 400 seconds, as shown in Fig. 4.11(d);
- **Scenario 5**: u_i's attention is on one position (e.g., Position A) for 400 seconds while u_j's attention periodically switches between Position A and one other position (e.g., Position B) every 10 seconds, as shown in Fig. 4.11(e);
- **Scenario 6**: u_i's attention is on one position (e.g., Position A) for 400 seconds while u_j's attention periodically switches between Position A and other positions (e.g., Position B, C, D, E) every 10 seconds, as shown in Fig. 4.11(f);

Performance Metrics

The performance metrics used to evaluate the performance of the proposed system are introduced as follows.

- **Sensitivity** = $\frac{TP}{FN+TP}$, which measures the system capability to correctly detect the users with the same visual attention.
- **Specificity** = $\frac{TN}{TN+FP}$, which measures the system capability to correctly detect the users with different visual attention.
- **Accuracy** = $\frac{TP+TN}{TP+TN+FP+FN}$, which measures the system capability to correctly detect the relationship between the visual attention of two users.

where TP, TN, FP, and FN represent the number of true positive results, the number of true negative results, the number of false positive results, and the number of false negative results, respectively. In our work, a true positive result is considered when the users with the same visual attention are detected as sharing the same visual attention, a true negative result indicates that the users with different visual attention are detected as having different visual attention, a false positive result signals that the users with different visual attention are detected as sharing the

same visual attention, and a false negative result means that the users with the same visual attention are detected as having different visual attention.

Analysis of Visual Attention Similarity

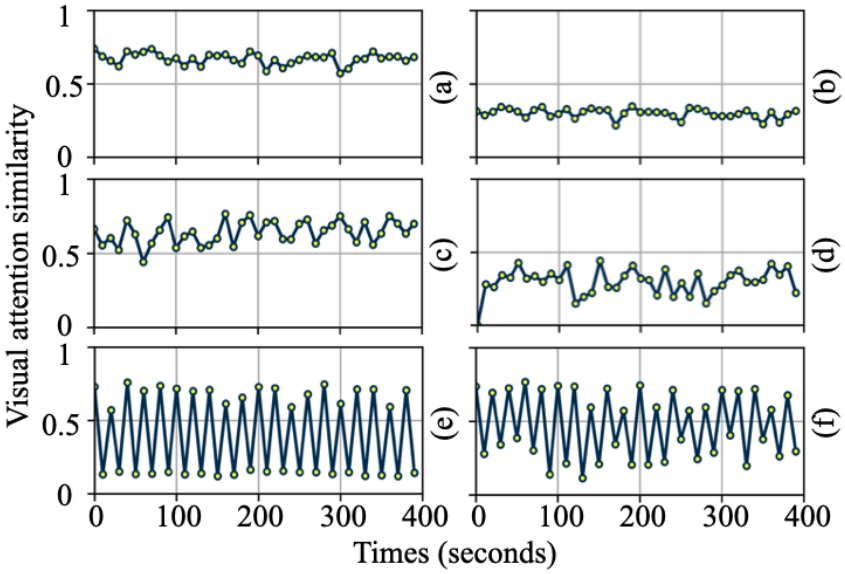


Figure 4.12: Visual attention similarity between u_i and u_j under different scenarios in the indoor office.

The experimental results in the indoor office are shown in Fig. 4.12, where Fig. 4.12(a) to Fig. 4.12(f) present the visual attention similarity between u_i and u_j in Scenario 1 to Scenario 6, respectively. As can be seen, the visual attention similarity in Fig. 4.12(a) and Fig. 4.12(c) are significantly larger than the visual attention similarity in Fig. 4.12(a) and Fig. 4.12(c). This is because u_i and u_j share the same attention in Scenario 1 and Scenario 3, while u_i's attention and u_j's attention are on different positions in Scenario 2 and Scenario 4. Moreover, more fluctuations are found in Scenario 3 and Scenario 4 because u_i's attention and u_j's attention keep changing in Scenario 3 and Scenario 4, whereas the attention of u_i and u_j remains unchanged in Scenario 1 and Scenario 2. We attribute such fluctuations to the changes in users' attention. Fig. 4.12(e) and Fig. 4.12(f) present the visual attention similarity in Scenario 5 and Scenario 6 where periodic changes are observed. This is because u_i's attention is always on one position, while u_j's attention periodically switches between this position and other positions in Scenario 5 and Scenario 6. More fluctuations are found in Fig. 4.12(f) than Fig. 4.12(e), because u_j's attention changes between only two fixed positions (e.g., Position A and Position B) in Scenario 5, whereas u_j's attention changes between Position A and any other positions in Scenario 6.

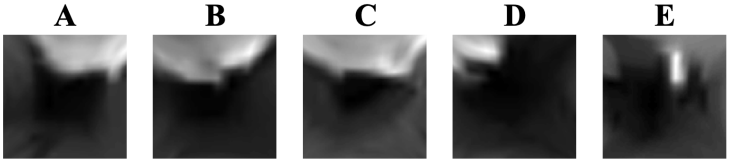
As a consequence, the proposed visual attention similarity is preliminarily observed as capable of distinguishing users' attention in static scenarios (Scenario 1 and Scenario 2), in dynamic scenarios (Scenario 3 and Scenario 4), as well as in hybrid scenarios (Scenario 5 and Scenario 6). Note that we only demonstrate the experimental results for the indoor office, since the curves of visual attention similarity for other environments exhibit similar patterns. A comprehensive comparison of system performance in different environments is given in the next section.

System Performance in Different Environments

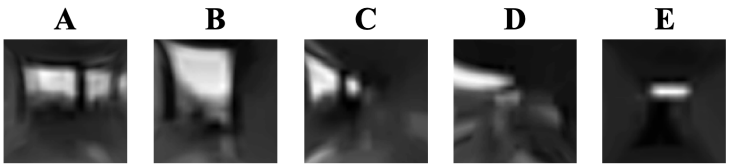
Table 4.1: System performance in different environments.

	Computer screen	**Indoor office**	**Outdoor campus**	**Overall**
Sensitivity	95.67 %	98.25 %	93.83 %	95.92 %
Specificity	94.58 %	97.83 %	88.00 %	93.47 %
Accuracy	95.13 %	98.04 %	90.92 %	94.69 %

Statistical analysis is given to comprehensively evaluate the system performance in different environments. As shown in Table 4.1, the overall sensitivity, specificity, and accuracy of the system are 95.92%, 93.47% and 94.96% respectively. While the best performance is found in the indoor office, where the sensitivity, specificity, and accuracy are 98.25%, 97.83%, and 98.04% respectively, the worst performance is found in the outdoor campus with sensitivity, specificity, and accuracy of 93.83%, 88.00%, and 90.92%, respectively. When users look at the computer screen, our system performs slightly worse than the indoor office but significantly better than the outdoor campus, with sensitivity, specificity, and accuracy of 95.67%, 94.58%, and 95.13% respectively.



(a) Light distributions on the eyes of users in the outdoor campus.



(b) Light distributions on the eyes of users in the indoor office.

Figure 4.13: Light distributions on the eyes of users.

The above results accord with the actual situations, i.e., the proposed system performs worst outdoors and best indoors. This is because the dominant light source outdoors is the natural light from the sky, which may not be occluded when the user is not surrounded by buildings. In this case, the light distribution in the user's eyes does not change significantly when his attention changes from one position to another. As a result, the proposed system may fail to detect changes in the user's attention in the outdoor environment. In contrast, indoor light sources are primarily from electric lights or outside windows, which are prone to be partially obscured and thus generate different light distributions in the user's eyes. Consequently, small changes in the user's attention may lead to significant changes in the light distribution in the user's eyes, and our system is more capable of distinguishing the users' attention in the indoor environment. As shown in Fig. 4.13(a), the images under B and C are highly similar, which represent the light distributions in the user's eyes when his attention is on Position B and Position C in the outdoor environment, respectively. Conversely, the light distributions are diverse in the indoor environment, as shown in Fig. 4.13(b).

System Performance under Diverse Scenarios

Further analysis is undertaken to study the system performance under different attention scenarios. As shown in Fig. 4.14(a), the overall performance is the best under static scenarios where the sensitivity, specificity, and accuracy are 99.33%, 94.75%, and 97.04%, respectively. The worst performance is found under dynamic scenarios with sensitivity, specificity, and accuracy of 92.67%, 91.75%, and 92.21%, respectively. The performance under hybrid scenarios is better than dynamic scenarios but worse than static scenarios, with sensitivity, specificity, and accuracy of 95.75%, 93.92%, and 94.83%, respectively. Similar results are also observed in every single environment. In the outdoor environment, system sensitivity, specificity, and accuracy significantly degrade from 98.50%, 87.75%, and 93.13% under static scenarios to 89.25%, 85.75%, and 87.50% under dynamic scenarios, as shown in Fig. 4.14(b). Similarly, when users look at computer screens, significant degradation of system sensitivity, specificity, and accuracy is observed from 99.75%, 98.00%, and 98.88% under static scenarios to 91.50%, 92.50%, and 92.00% under dynamic scenarios, as shown in Fig. 4.14(c). A slight degradation in system performance is found in the indoor environment, where the system sensitivity, specificity, and accuracy drop from 99.75%, 98.50%, and 99.13% under static scenarios to 97.25%, 97.00%, and 97.13% under dynamic scenarios, as shown in Fig. 4.14(d).

These results are in accordance with the actual situation. Because the users' attention remains unchanged under static scenarios, the system is more capable of correctly detecting the relationship between users' attention. On the contrary, users' attention keeps changing under dynamic scenarios, which renders the system more difficult to correctly detect the relationship between users' attention. However, the proposed system achieves sensitivity, specificity, and accuracy of 89.25%,

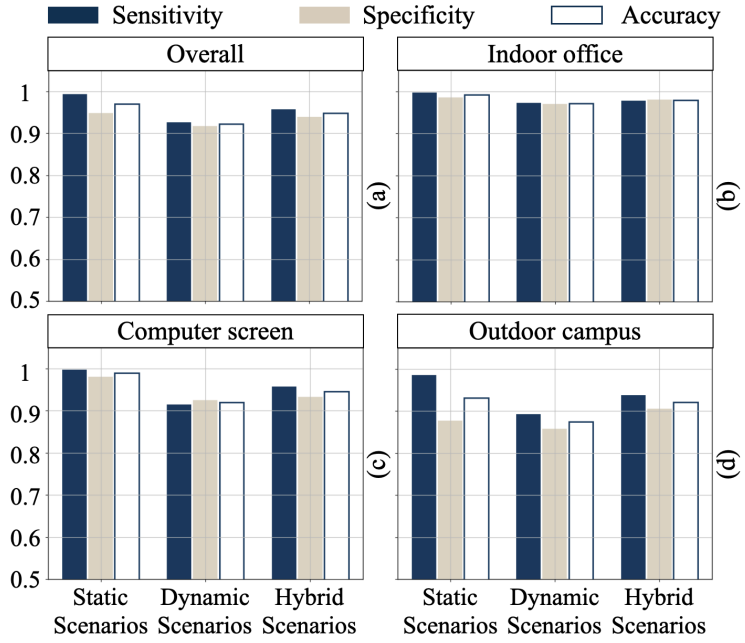


Figure 4.14: System performance in different scenarios.

85.75%, and 87.50% even in the worst case, i.e., in the outdoor environment under dynamic scenarios, as shown in Fig. 4.14(d). This demonstrates the robustness and adaptability of the proposed system to different environments and scenarios.

System Performance with Various External Factors

Table 4.2: System performance with various external factors

	Indoor light on	**Sunlight shining**	**Background changing**
Sensitivity	82.17 %	70.67%	93.33 %
Specificity	78.42 %	63.00 %	91.42 %
Accuracy	80.29%	66.83%	92.38 %

This section investigates the impact of external factors on system performance, i.e., the background of the computer screen, the room lights, and the sunlight. To achieve this, 3 groups of experiments are conducted where the experimental environment is the computer screen, as shown in Fig. 4.10(b), and the experimental scenarios are the 6 scenarios described in Fig. 4.11. In each group of experiments, the user u_i looks at a screen with a fixed background in an office without room light and sunlight. In the first group of experiments, u_j looks at a screen with changing backgrounds in an office without room light and sunlight. In the second group of

experiments, u_j looks at a screen with a fixed background in an office with room light and without sunlight. In the third group of experiments, u_j looks at a screen with a fixed background in an office without room light and with sunlight.

Table 4.2 and Fig. 4.15(a) jointly present the statistical results. As shown in Table 4.2, changing the background of the computer screen does not impact system performance, where the overall sensitivity, specificity, and accuracy achieve 93.33%, 91.42%, and 92.38%. In this case, the system performs satisfactorily under every scenario, with sensitivity, specificity, and accuracy of 97.00%, 96.00%, and 96.50% under static scenarios, 90.25%, 86.75%, and 88.50% under dynamic scenarios, and 92.75%, 91.50%, and 92.13% under hybrid scenarios, as shown in Fig. 4.15(a). However, the system performance drops significantly when the room lights are turned on, with overall sensitivity, specificity, and accuracy of 82.17%, 78.42%, and 80.29%, as shown in Table 4.2. The system performs unsatisfactorily under every scenario, with sensitivity, specificity, and accuracy of 87.00%, 78.25%, and 82.63% under static scenarios, 77.25%, 75.25%, and 76.25% under dynamic scenarios, and 82.25%, 81.75%, and 82.00% under hybrid scenarios, as shown in Fig. 4.15(b). More remarkably, when sunlight is allowed into the room, the proposed system is severely impaired, with overall sensitivity, specificity, and accuracy of only 70.67%, 63.00%, and 66.83%, as shown in Table 4.2. The system is incapable of detecting the relationship between users' attention under every scenario, with sensitivity, specificity, and accuracy of only 72.75%, 65.50%, and 69.13% in static scenarios, only, 69.00%, 59.75%, and 64.38% in dynamic scenarios, and only 70.25%, 63.75%, and 67.00% in hybrid scenarios, as shown in Fig. 4.15(c).

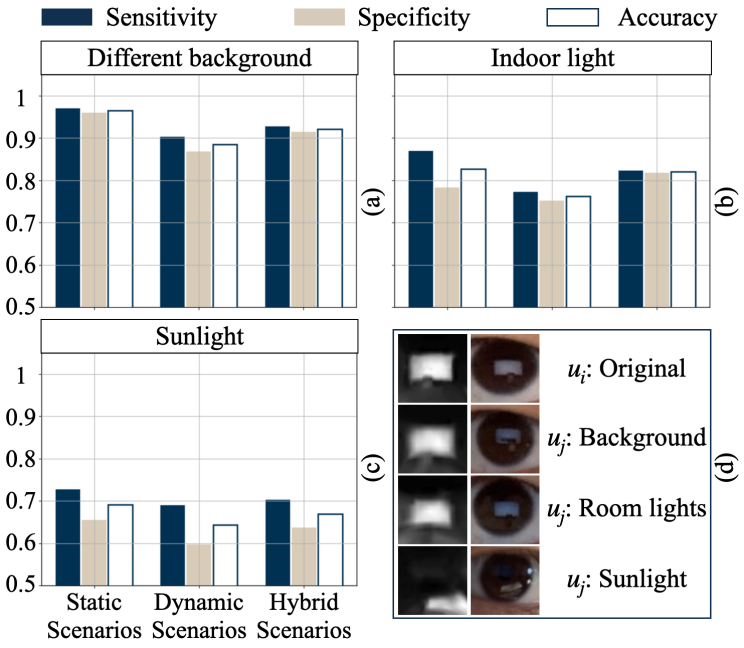


Figure 4.15: System performance with different external factors and scenarios.

We analyze the above results through Fig. 4.15(d), which shows from top to bottom the light distributions in the users' eyes 1) without any external factors, 2) with only the screen background changing, 3) with only the room lights on, and 4) with only sunlight in the room, respectively. As shown in Fig. 4.15(d), when the screen background changes, the light distribution of u_j shows little difference compared to the original light distribution of u_i, and thus the impact on system performance is minor. When the room lights are turned on, the light distribution u_j is significantly blurred compared to the original light distribution u_i. This is because the screen brightness is relatively reduced as the room lights increase the room brightness, which reduces the light amount reflected on u_j's eyes, thereby impacting the system performance. When sunlight falls on the table in front of the computer screen, it takes over as the dominant light source because sunlight is much brighter than the screen. As a result, u_i and u_j exhibit completely different light distributions although they share similar attention, and thus a significant impairment is found in system performance.

Performance Comparison with other Research

Table 4.3: Performance comparison with other research.

	TMK&PDQ +PDQF	HOOF-based approach	Visil	Our approach
Sensitivity	87.64%	81.56%	81.30%	95.98%
Specificity	81.04%	76.64%	75.76%	88.52%
Accuracy	84.34%	79.10%	78.53%	92.25%

In this section, we compare the performance of the proposed approach with the other three approaches introduced in the related work section, i.e., the hashing-based approach [DWB19], the feature-based approach [RCW+12], and the deep learning-based approach [KPP+19]. As shown in Table 4.3, the hashing-based approach exhibits a significant degradation compared with our approach, with sensitivity, specificity, and accuracy of 87.64%, 81.04%, and 84.34%. In addition, a more significant drop in performance is observed for the feature-based approach in Table 4.3, with sensitivity, specificity, and accuracy of 81.56%, 76.64%, and 79.10% respectively. Remarkably, the deep learning-based approach severely impairs system performance compared with our approach, with sensitivity, specificity, and accuracy of 81.30%, 75.76%, and 78.53% respectively.

We attribute the degradation in the hashing-based approach to the limited color information in the video extracted from human eyes, where the primary color components are a large area of dark background and a small area of bright foreground. Therefore, the useful information for the bright foreground is very sparse in the hash codes generated for the videos. The drop in the feature-based approach is attributed to the singularity of the extracted features, which are mostly the edges

between the dark background and the bright foreground. The similarity between videos is hardly quantified based on only edge features. For the deep learning-based approach, the features from the convolutional layers are leveraged and aggregated as the representations of videos. However, due to the low resolution of the videos in our case, the pre-trained CNN hardly detects the objects in the videos and hence the features generated in each convolutional layer are unreliable.

4.6.2 Advanced Analysis based on Uncontrolled Experiments

In this section, a large-scale dataset is first established, and diverse attention traces are generated for users based on the collected dataset. The robustness of the proposed system is then studied by investigating the synchronicity between the relationship of users' attention traces and their visual attention similarity. A robust system is expected to be able to detect a higher visual attention similarity when the attention traces of users are highly overlapping and vice versa. Moreover, the effects of the importance level and equality level on detecting the visual similarity between users are also analyzed based on the generated attention traces.

Dataset Collection

A large-scale dataset is established in this section, which is collected by two users at 60 different positions in the university. These 60 positions, ranging from outdoors on the university campus, indoors in the university buildings, and different positions on the computer screen, provide sufficient support to verify the robustness of our system. At each position, two users respectively take a 1-minute video containing their eyes, so the entire dataset contains 2×60 videos and 2×60 labels indicating the actual positions of users. Fig. 4.16 shows the large-scale dataset, where the RGB images are the scenes users look at, and the grey-scale images are the corresponding light distributions in users' eyes.

Creation of Attention Traces

Based on the collected large-scale dataset, we generate a wide variety of attention traces for users u_i and u_j, as shown in Fig. 4.17. First, we recursively select the videos collected by u_j in the indoor buildings. The length of the video selected each time is random until the total length of all selected videos reaches 1 minute, which is the generated attention trace for u_j. Then, we generate u_i's attention trace by selecting the videos collected by u_i in the indoor buildings so that u_i's attention trace and u_j's attention trace are 100% overlapping. Next, u_j's attention trace is kept unchanged and a segment of u_i's attention is randomly changed so that u_i's attention trace and u_j's attention trace are 90% overlapping. In this way, the overlap ratio of the attention traces between u_i and u_j is reduced by 10% for each time, until the overlap ratio of the attention traces between u_i and u_j is 0%. These steps

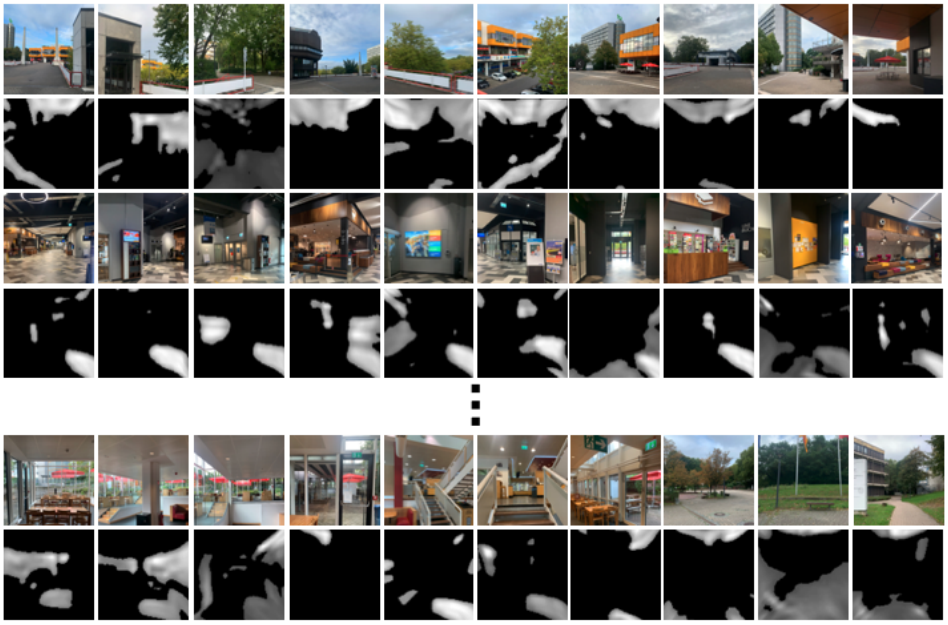


Figure 4.16: Large-scale dataset.

are repeated 100 times to produce 1000 combinations of attention traces between u_i and u_j in indoor buildings.

In addition to the videos collected indoors in the university buildings, the attention traces are also generated for u_i and u_j using the videos collected outdoors on the university campus, and the videos collected when u_i and u_j look at the computer screen. The proposed system then computes the visual attention similarity between u_i and u_j for each pair of attention traces to obtain large-scale statistical results, which are presented in the next section.

Synchronicity Analysis

The experimental results are shown in Fig. 4.18, where the x-axis in each figure in Fig. 4.18 is the overlap ratio of users' attention traces, and the y-axis is the visual attention similarity between users detected by the proposed system. Each curve in Fig. 4.18 consists of 11 bars, which represent the distributions of visual attention similarity between users at 11 different overlap ratios, i.e., 0%, 10%,...,100%. The midpoint of each bar indicates the mean of visual attention similarity at a given overlap ratio, while the length of the bar represents the standard deviation of visual attention similarity. Therefore, Fig. 4.18 investigates the synchronicity between the overlap ratio of users' attention traces and their visual attention similarity in different environments, where the red curve in Fig. 4.18(a) represents indoors the university buildings, the yellow curve in Fig. 4.18(b) represents the computer screens and the blue curve in Fig. 4.18(c) stands for outdoors the university campus. For

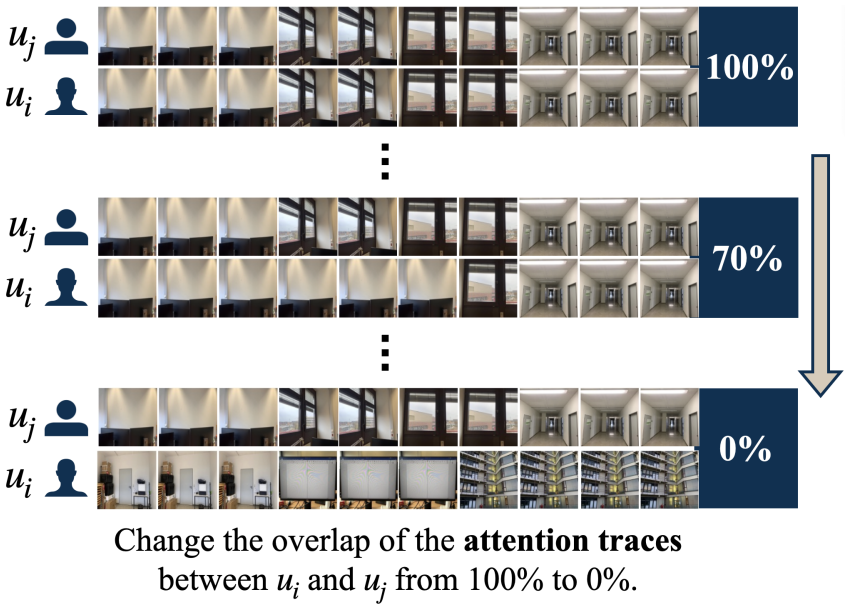
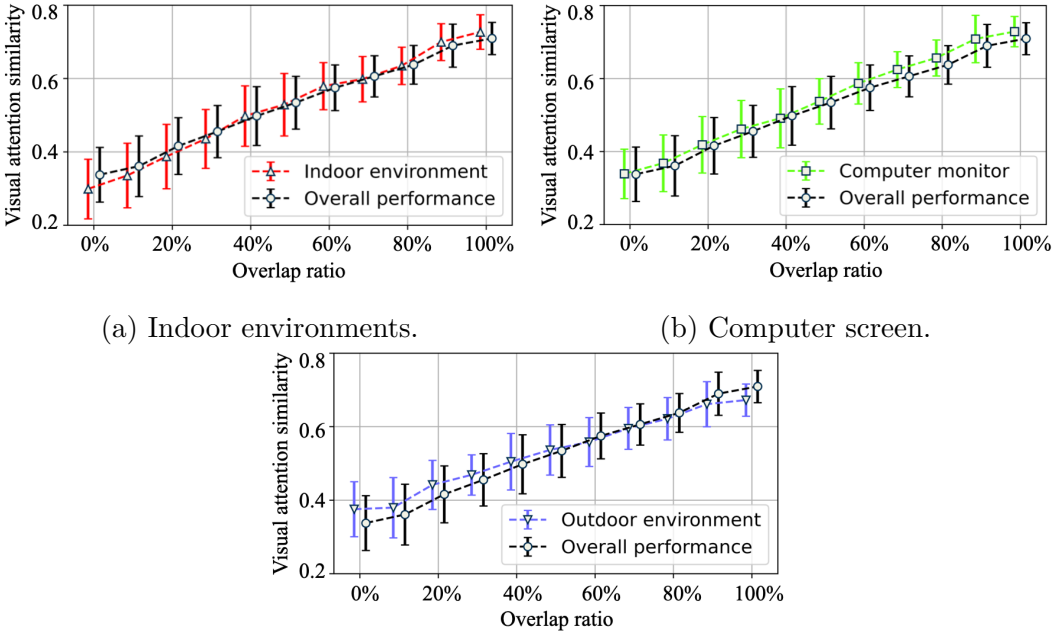


Figure 4.17: Creation of attention traces.

the purpose of comparison, the overall mean and overall standard deviation of visual attention similarity are presented by black curves in each figure in Fig. 4.18.

We can see that each curve in Fig. 4.18 exhibits a similar trend, that is, the mean of visual attention similarity increases as the overlap ratio of users' attention traces increases, and the standard deviation of visual attention similarity stabilizes at a certain range despite the increase of the overlap ratio of users' attention traces. An example is given by the red curve in Fig. 4.18(a), where the mean of visual attention similarity increases from 0.3 to 0.5 and eventually to 0.72, as the overlap ratio of users' attention traces increase from 0% to 50% and finally to 100%. The standard deviation of visual attention similarity is 0.18, 0.19, and 0.11, when the overlap ratio of users' attention traces increases from 0% to 50% and finally to 100%, respectively. Moreover, we can see that the slope of the red curve in Fig. 4.18(a) is the largest, while the slope of the blue curve in Fig. 4.18(c) is the smallest. This shows that our system's ability to distinguish users' attention is best in the indoor environment and worst in the outdoor environment, which is also in accordance with our analysis in Section 4.6.1. Note that even in the worst outdoor environment, our system is able to satisfactorily distinguish users' attention. As shown by the blue curve in Fig. 4.18(c), while the overlap ratio of users' attention traces increases from 0% to 50% and finally to 100%, the mean of visual attention similarity increases from 0.3 to 0.5 and eventually to 0.72, which is distinguishable in the outdoor environment.



(a) Indoor environments.

(b) Computer screen.

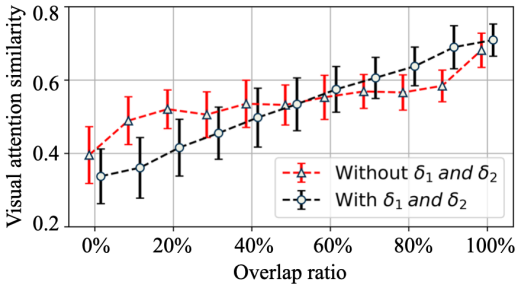
(c) Outdoor environments.

Figure 4.18: Correlation between visual attention similarity and overlap of attention traces.

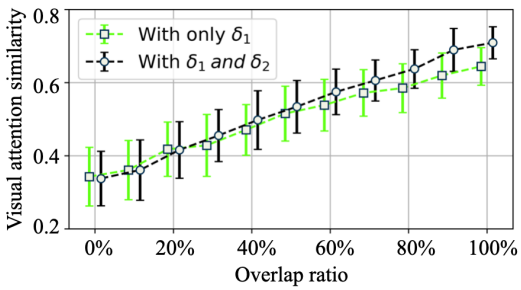
Impact of Parameters

The impact of the proposed importance level and equality level are investigated in this section. To achieve this, the visual attention similarity is re-computed based on the uncontrolled experiments 1) when neither the importance level δ_1 nor the equality level δ_2 is considered, 2) when only the importance level δ_1 is considered, and 3) when only the equality level δ_2 is considered, respectively. The experimental results are shown by the red curve in Fig. 4.19(a), the yellow curve in Fig. 4.19(b), and the blue curve in Fig. 4.19, respectively. For the purpose of comparison, the black curve in each figure in Fig. 4.19 represents the results when considering both the importance level δ_1 and the equality level δ_2. Similar to Fig. 4.18, all curves in Fig. 4.19 also describe the synchronicity between the overlap ratio of users' attention traces and their visual attention similarity.

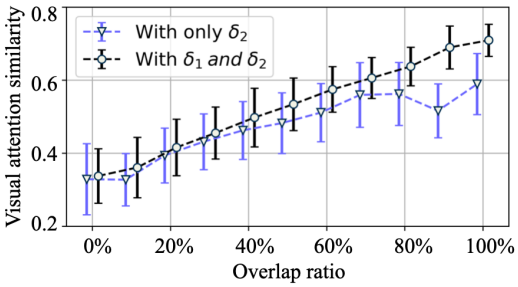
The red curve is much flatter compared to the black curve in Fig. 4.19(a), with the visual attention similarity increasing from 0.48 only to 0.59 when the overlap ratio of users' attention traces increases from 10% to 90%. The significant difference in visual attention similarity is only found when the overlap ratio is 0% or 100%, i.e., when users' attention traces are completely identical or different. Therefore, when neither the importance level δ_1 nor the equality level δ_2 is considered, the proposed system fails to correctly detect users' attention. Both the yellow curve in Fig. 4.19(b) and the blue curve in Fig. 4.19(c) are very similar to the black curve



(a) Without importance level and equality level.



(b) With only importance level.



(c) With only equality level.

Figure 4.19: Impact of importance level and equality level.

when the overlap ratio of users' traces is from 0% to 60%, and both of them are slightly lower than the black curve when the overlap ratio is from 60% to 100%. As a consequence, the ability of the proposed system in distinguishing users' attention is significantly improved, when either the importance level or the equality level is taken into account. However, the best performance of our system is only achieved when both the importance level and the equality level are exploited.

4.7 Conclusion

This work designs a visual attention correlating system based on the collaboration between cross-domain sensing data, i.e., the movements of human eyeballs and the light reflected in human eyes. The designed system first creates consecutive visual signatures for each user as his visual trajectory based on the collected eyeballs' movements and the light patterns. A DTW-based algorithm is proposed in the system to align users' visual signatures. The maximum similarity between users' visual signatures is referred to as the visual attention similarity between users. Extensive experiments demonstrate that the proposed system is able to distinguish users' visual attention and is robust to different environments, scenarios, and external conditions.

Chapter 5

Multi-Model-Based Generation of Sensing Data

5.1 System Model

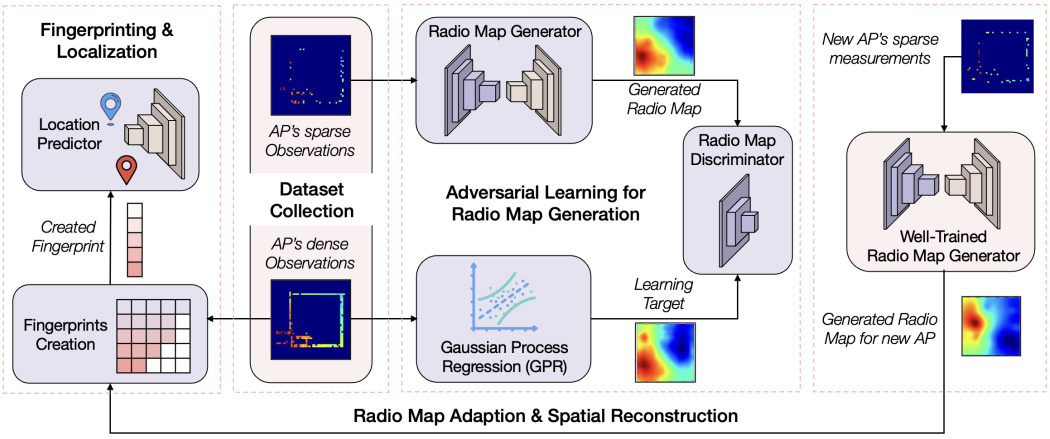


Figure 5.1: Overview of system framework.

5.1.1 System Modelling

Given an indoor environment divided into $h \times w$ discrete grids, let $\hat{\mathbf{R}} = \{\mathbf{r_i} | 1 \leq i \leq h \times w\}$ denote the set of two-dimensional (2-D) coordinates of these grids. Initially, I_1 reference points (RPs) $\mathbf{R} = \{\mathbf{r}_i | 1 \leq i \leq I_1\}$ are selected from these grids $\hat{\mathbf{R}}$, where $\mathbf{r}_i = (x_i, y_i)$ is the 2-D coordinate of the i-th RP. At each RP, J_1 Wi-Fi samples are offline collected, where a Wi-Fi sample refers to a 2-tuples list, including the MAC addresses of detected Wi-Fi access points (APs) and their corresponding RSS. First, let $\mathbb{A} = \{\alpha_{i,j} | i \in [0, I_1], j \in [0, J_1]\}$ denote the set of $I_1 \times J_1$ Wi-Fi samples collected offline, where $\alpha_{i,j}$ is the j-th Wi-Fi sample collected at $\mathbf{r}_i$. Assume that K_1 APs are observed during the offline collection phase. Second, let $\mathbb{D} = \{d_{j,k} | j \in [0, J_1], k \in [0, K_1]\}$ denote a set of $N \times N$ images ($N > h, w$) representing the RSS of each AP at all I_1 RPs, where the image intensity $d_{j,k}(x_i, y_i)$ indicates the RSS of AP-k observed in $\alpha_{i,j}$. Note that the pixels outside the RPs of each image are set to zero. Third, I_2 RPs $\widetilde{\mathbf{R}} = \{\widetilde{\mathbf{r}}_i | 1 \leq i \leq I_2\}$ are sparsely sampled from $\mathbf{R}$ ($I_2 \ll I_1$) and let $\mathbb{S} = \{s_{j,k} | j \in [0, J], k \in [0, K_1]\}$ denote another set of $N \times N$

images representing the RSS value of each AP at these I_2 RPs. In this work, $\mathbb{D}$ and $\mathbb{S}$ are referred to as the dense and sparse radio maps of Wi-Fi APs. In addition, full radio maps of AP-k refer to that AP-k has valid RSS values at all grids $\hat{\mathbf{R}}$ in the environment.

Given $\mathbb{S}$ and $\mathbb{D}$, this work models the propagation behavior of APs' Wi-Fi signals in the target environment, by learning the sparse-to-dense evolution of radio maps of these APs. To this end, this work adversarially trains a generator and a discriminator for radio map adaptation. On one hand, we assume that new APs are observed during the online stage, and their RSS values are sparsely collected. The well-trained generator is capable of generating full radio maps for these APs using their sparse RSS observations, i.e., generating RSS for them at each grid in the discrete environment. On the other hand, this generator can also be employed to densify the radio maps of existing APs, thereby alleviating the efforts required for offline collection. Given $\mathbb{A}$, an auxiliary objective of this work is to devise a novel fingerprinting technique for localization. This fingerprinting technique progressively explores the RSS correlation between APs, and exhibits its ability against performance degradation caused by signal fading.

5.1.2 System Overview

Fig. 5.1 presents an overview of the system framework, comprising four phases: *Phase 1*) Dataset collection, *Phase 2*) Adversarial learning-based radio map generation, *Phase 3*) RSS correlation-based fingerprinting and localization, and *Phase 4*) Radio map adaptation and spatial reconstruction.

In Phase 1, site surveys collect Wi-Fi samples from ambient Wi-Fi APs at dense RPs. The collected Wi-Fi samples are first transformed into dense and sparse radio maps of Wi-Fi APs. Next, the dense and sparse radio maps of Wi-Fi APs are forwarded to Phase 2, while the collected raw Wi-Fi samples are fed into Phase 3. In Phase 2, a framework for radio map generation incorporating GAN and GPR is proposed. The generator in this GAN-GPR framework learns the sparse-to-dense evolution of radio maps of Wi-Fi APs, aiming at generating full radio maps for Wi-Fi APs. In Phase 3, a novel fingerprinting technique is devised to transform the collected Wi-Fi samples into fingerprints. The devised fingerprinting technique creates a fingerprint dataset by studying the RSS correlation between APs detected in Phase 1. After that, a location predictor is trained for indoor localization using the created fingerprint dataset. Radio maps adaptation occurs in Phase 4, where the generator trained in Phase 3 not only generates radio maps for newly observed APs but also densifies the radio maps of existing APs with RSS generation. Additionally, the location predictor is retrained using the adapted radio maps in Phase 4.

5.2 Adversarial Learning for Radio Map Generation

To model the propagation of APs' wireless signals in the target environment and generate full radio maps for APs, this work designs a GAN-GPR framework to learn the sparse-to-dense evolution of APs' radio maps. The proposed GAN-GPR framework is composed of a radio map generator, a discriminator, and a GPR model. The generator takes sparse radio maps of Wi-Fi APs as network input and aims to generate full radio maps for APs. In parallel, the GPR model creates full radio maps for APs based on their dense radio maps, which serve as the learning target for the generator. The discriminator is adversarially trained to separate generator-derived full radio maps from GPR-derived full radio maps, until the generator is capable of generating radio maps to fool the discriminator. In this chapter, the technical details of the proposed framework are explained.

5.2.1 Framework Design

Given the dense radio maps $\{d_{j,k}|j \in [1, J_1]\}$ of AP-k, the RPs' coordinates $\mathbf{R}$ where $d_{j,k}$ are collected, and the coordinate $\hat{\mathbf{R}}$ of all grids in the environment, the proposed framework first devises a GPR model to estimate AP-k's RSS at $\hat{\mathbf{R}}$ by taking $\mathbf{R}$ and $d_{j,k}$ as the training data. Let c_k denote the estimated RSS of AP-k at $\hat{\mathbf{R}}$, which is also referred to as the full radio maps of AP-k as $\hat{\mathbf{R}}$ includes all grids in the environment. Eq. (5.1) demonstrates how the full radio map c_k of AP-k is created, which is essentially to find the posterior probability of the designed GPR model. In Eq. (5.1), $\mathcal{K}(\mathbf{R}, \mathbf{R})$ explores the covariance between the training points, $\mathcal{K}(\hat{\mathbf{R}}, \mathbf{R})$ studies the covariance between the training points and each test point, and σ_n^2 is the noise. To address the problem that RSS of AP-k varies over time due to signal fading, the dense radio maps are averaged and denoted by $\frac{\sum_{j=1}^{J_1} d_{j,k}}{J_1}$ in Eq. (5.1).

$$c_k = \mathcal{K}(\hat{\mathbf{R}}, \mathbf{R}) \cdot (\mathcal{K}(\mathbf{R}, \mathbf{R}) + \sigma_n^2 \cdot \mathbf{I})^{-1} \cdot \frac{\sum_{j=1}^{J_1} d_{j,k}}{J_1}. \tag{5.1}$$

The covariance function employed in the GPR model comprises two radial basis function (RBF) kernels with different length scales ℓ_1 and ℓ_2 ($\ell_1 \gg \ell_2$), as shown in Eq. (5.2). The kernel with the larger length scale depicts the coarse distribution of APs' RSS in the environment, while the kernel with the smaller length scale preserves the RSS distribution details from the training data.

$$\mathcal{K}(\mathbf{r}, \mathbf{r}\prime) = \alpha \cdot \exp\left(-\frac{\|\mathbf{r} - \mathbf{r}\prime\|_2^2}{2\ell_1^2}\right) + \beta \cdot \exp\left(-\frac{\|\mathbf{r} - \mathbf{r}\prime\|_2^2}{2\ell_2^2}\right), \quad \mathbf{r}, \mathbf{r}\prime \in \mathbf{R}. \tag{5.2}$$

Taking the GPR-derived full radio map c_k as the learning target and the sparse radio map $s_{j,k}$ as network input, the proposed framework then trains a generator $\mathcal{G}$

for radio map generation. The well-trained generator $\mathcal{G}$ only utilizes AP-k's sparse radio map $s_{j,k}$ to generate the full radio map for AP-k, and the generator-derived full radio map $\mathcal{G}(s_{j,k})$ is expected to closely resemble the GPR-derived full radio map c_k.

$$\min_G \max_D \mathcal{L}(\mathcal{G},\mathcal{D}) = \mathbb{E}_{c_k\sim\mathbb{C}}[\mathcal{D}(c_k)] - \mathbb{E}_{s_{j,k}\sim\mathbb{S}}[\mathcal{D}(\mathcal{G}(s_{j,k}))] + \lambda\,\mathbb{E}_{p\sim\mathbb{P}}\left[(\|\nabla_p\mathcal{D}(p)\|_2 - 1)^2\right] \tag{5.3}$$

In the proposed framework, the generator $\mathcal{G}$ is trained in an adversarial manner with a discriminator $\mathcal{D}$, and the training process is essentially a min-max game according to Eq. (5.3). On one hand, the discriminator $\mathcal{D}$ is trained to distinguish between the GPR-derived full radio map c_k and the generator-derived full radio map $\mathcal{G}(s_{j,k})$. On the other hand, the generator $\mathcal{G}$ is trained to generate $\mathcal{G}(s_{j,k})$ that can fool $\mathcal{D}$. As shown by the loss function in Eq. (5.3), the objective of training the discriminator $\mathcal{D}$ is to maximize $\mathcal{L}(\mathcal{G},\mathcal{D})$, which is achieved by maximizing $\mathbb{E}_{c_k\sim\mathbb{C}}[\mathcal{D}(c_k)]$, i.e., giving a higher score when $\mathcal{D}$ detects c_k. In contrast, the objective of training the generator $\mathcal{G}$ is to minimize $\mathcal{L}(\mathcal{G},\mathcal{D})$ by maximizing $\mathbb{E}_{s_{j,k}\sim\mathbb{S}}[\mathcal{D}(\mathcal{G}(s_{j,k}))]$, i.e., generate the full radio map to fool $\mathcal{D}$. Note that, to avoid violating the Lipschitz continuity constraint during training, i.e., an infinitely large (infinitely small) score is directly given when $\mathcal{G}$ detects c_k ($\mathcal{G}(s_{j,k})$), the gradient penalty proposed in [GAA+17] is applied during training, denoted by $(\|\nabla_p\mathcal{D}(p)\|_2 - 1)^2$. The key idea of the gradient penalty is to keep the gradient norm $\|\nabla_p\mathcal{D}(p)\|_2$ close to 1 for any given point p, which is sampled from the interpolated points set $\mathbb{P} = \eta\cdot c_k + (1-\eta)\cdot G(s_{j,k})$.

The proposed framework for radio map generation incorporates the advantages of both GPR and GAN. First, when an AP only provides extremely sparse RSS observations, the kernel function of GPR is incapable of capturing the spatial correlation between data points. In this case, the trained generator can generate full radio maps for APs using their sparse RSS observations, because the signal propagation in the target environment is studied by learning the sparse-to-dense evolution of APs' radio maps. Second, GPR significantly outperforms GAN-based networks in terms of interpolation capability. Therefore, the proposed framework can effortlessly generate denser radio maps for APs.

5.2.2 Network Architecture

Fig. 5.2 presents the network architectures of the designed generator $\mathcal{G}$ and discriminator $\mathcal{D}$, where the generator design adopts the concept of the U-net network [RFB15] for better capturing the fine-grained features from the radio maps. First, a set of $N \times N$ sparse radio maps are fed into the generator at the input layer. Then, the sparse radio maps are downsampled to preserve only the most important features, through a combination of a series of convolutional layers, norm layers, and activation layers. Based on the retained features, another combination of convolutional layers,

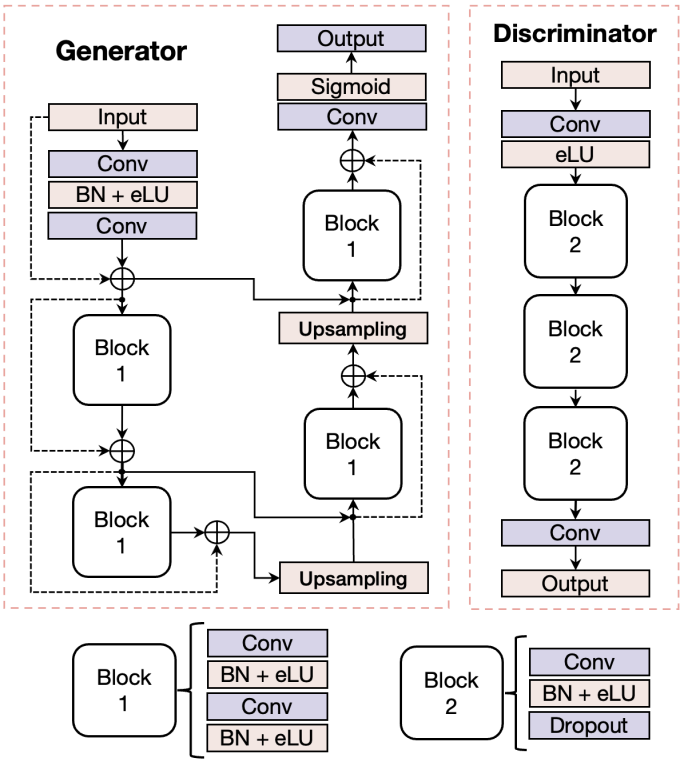


Figure 5.2: Network architecture.

norm layers, and activation layers is employed to upsample the downsampled radio maps. The upsampled $N \times N$ images represent the full radio maps to be generated. Note that, the Resnet architecture [HZR+16] in the generator network optimizes the training of the generator by associating the inputs and outputs of each block. In addition, the inputs of each block in downsampling are fed into the inputs of each block in upsampling, for the purpose of better retaining spatial features.

The architecture of the discriminator network in Fig. 5.2 is simpler than the generator network. The discriminator simultaneously takes the generator-derived full radio maps and the GPR-derived full radio maps as network inputs. The input dimension is constantly reduced through the combination of multiple convolutional layers, norm layers, and activation layers, until ultimately obtaining a 1×1 score which quantifies the generation quality.

5.3 RSS Correlation-based Fingerprinting and Localization

5.3.1 Creation of Fingerprints

Given the set of all $I_1 \times J_1$ Wi-Fi samples, denoted by $\mathbb{A} = \{\alpha_{i,j} | i \in [0, I_1], j \in [0, J_1]\}$, collected during an offline collection phase, the fingerprint dataset $\mathbb{F} = \{f_{i,j} | i \in$

Algorithm 4: RSS Correlation-based Fingerprinting.

Input : Ω: A list keeping the alphabetically sorted BSSIDs of all APs observed during the dataset collection phase;
$\alpha_{i,j}$: The j-th Wi-Fi samples collected at the (x_i, y_i);
Output : $f_{i,j}$: The fingerprint created for $\alpha_{i,j}$

1 //**Initialization:**
2 A list $\gamma_{i,j}$ of length $|\alpha_{i,j}|$ is initialized for keeping the importance level of each AP in $\alpha_{i,j}$.
3 A list $f_{i,j}$ of length $|\Omega|$ is initialized for keeping the fingerprint.
4 //**Computation of importance level:**
5 **for** q *from* 1 *to* $|\alpha_{i,j}|$ **do**
6 **for** p *from* 1 *to* $|\alpha_{i,j}|$ **do**
7 $l \leftarrow 1$
8 **Function** `AP_Correlation`(l, r_q, r_p)**:**
9 **if** $r_q - r_p < \delta \cdot l$ **then**
10 **return** $\gamma_{i,j}(q)$
11 $\gamma_{i,j}(q) \leftarrow \gamma_{i,j}(q) + l$
12 $l \leftarrow l + 1$
13 **return** `AP_Correlation`(l, r_q, r_p)
14 //**Creation of fingerprint:**
15 **for** q *from* 1 *to* $|\alpha_{i,j}|$ **do**
16 **for** p *from* 1 *to* $|\Omega|$ **do**
17 **if** $w_q = \Omega_p$ **then**
18 $f_{i,j}(p) \leftarrow \gamma_{i,j}(q)$
19 //**Normalization:** $f_{i,j} \leftarrow f_{i,j}/M$;
20 **return** $f_{i,j}$;

$[0, I_1], j \in [0, J_1]\}$ is created using the novel fingerprinting technique $f_{i,j} = \mathcal{F}(\alpha_{i,j})$ devised in this work. The design rationale of the proposed fingerprinting technique is to progressively investigate the RSS correlation between each pair of APs in $\alpha_{i,j}$, and then quantify the importance level of each AP in $\alpha_{i,j}$. More specifically, if the RSS of a given AP is significantly stronger than the RSS of other APs in $\alpha_{i,j}$, the importance level of this AP is considered higher than that of other APs in $\alpha_{i,j}$. As the importance level of each AP in $\alpha_{i,j}$ is jointly determined by all observed APs, even if the RSS of a particular AP undergoes an abrupt change due to signal fading indoors, the importance level is not impacted significantly. The creation of the fingerprint $f_{i,j}$ for $\alpha_{i,j}$ is essentially to compute the importance levels of each observed AP in $\alpha_{i,j}$.

Algorithm 4 illustrates the technical details of the devised fingerprinting technique using the example of generating the fingerprint $f_{i,j}$ for the Wi-Fi sample $\alpha_{i,j}$. Here, $\alpha_{i,j} = \{(w_1, r_1), (w_2, r_2), ..., (w_q, r_q), ...)\}$ is a 2-tuples list where w_q represents the MAC address of the q-th AP in $\alpha_{i,j}$ and r_q is the RSS of w_q. To compute the importance level of w_q in $\alpha_{i,j}$, Algorithm 4 investigates the RSS correlation between w_q and each AP in $\alpha_{i,j}$. As shown in lines 5 to 13 in Algorithm 4, a recursive function *AP_Correlation* is executed to study the correlation between w_q's RSS

(r_q) and w_p's RSS (r_p). As can be seen, the importance level of w_q, denoted by $\gamma_{i,j}(q)$, increases by l if r_q is larger than r_p by $l \cdot \delta$. Next, the value of l is incremented by 1, and the recursive function proceeds. In the next recursion, if r_q is larger than r_p by $l \cdot \delta$, $\gamma_{i,j}(q)$ increases by l and the value of l is incremented by 1 again. When l increases until r_q is no longer larger than r_p by $l \cdot \delta$, the recursive function terminates, and w_q is compared with the next AP w_{p+1} in $\alpha_{i,j}$. The above procedure is iterated until w_q is compared with each AP in $\alpha_{i,j}$, and $\gamma_{i,j}(q)$ is computed. Similarly, the importance level of each AP in $\alpha_{i,j}$ can be computed using the above method. Finally, the importance level of each AP in $\alpha_{i,j}$ is filled into a new list $f_{i,j}$ according to lines 14 to 18 in Algorithm 4, where all APs observed during the offline collection phase are sorted alphabetically. Here, $f_{i,j}$ is the fingerprint created for $a_{i,j}$.

Fig. 5.3 gives a numerical example of creating the fingerprint for $\alpha_{i,j}$. Let us assume that $\alpha_{i,j}$ = {(AP1,-36dB), (AP2,-45dB), (AP3,-55dB), (AP4,-63dB), (AP5,-77dB), (AP6,-85dB)}, and δ is set to 8. The importance level of AP1 can be computed by exploring the correlation between the RSS of AP1 (r_1) and the RSS of other APs (r_2 to r_6) in $\alpha_{i,j}$. We first compare r_1 and r_6 by executing the recursive function *AP_Correlation*. The importance level of AP1 $\gamma_{i,j}(1)$ is initialized to 0, and l is initialized to 1. As can be seen, r_1 is greater than r_6 by 49 dB, which is larger than the value of 1δ. Therefore, $\gamma_{i,j}(1)$ is added by 1, l is added by 1, and the recursive function proceeds. Next, $\gamma_{i,j}(1)$ is added by 2 because 49 dB is still greater than the value of 2δ. Similarly, by recursively repeating the above procedures, $\gamma_{i,j}(1)$ is added by 3, 4, 5, and 6 as l is incremented to 3, 4, 5, and 6, respectively. The recursive function terminates when r_1 is not larger than r_6 by $l \cdot \delta$, i.e., 49 dB is smaller than 7δ. After compared with w_6, the importance level of w_1 is $\gamma_{i,j}(1) = 1+2+3+4+5+6 = 21$. Subsequently, w_1 is compared with w_5. When l increases from 1 to 5, $\gamma_{i,j}(1)$ is respectively added by 1, 2, 3, 4, and 5 because r_1 is larger than r_5 by 1δ, 2δ, 3δ, 4δ, 5δ. The recursive function terminates when l increases to 6, where r_1 is not larger than r_5 by 6δ. As a result, $\gamma_{i,j}(1) = 21 + 1 + 2 + 3 + 4 + 5 = 36$ after w_1 is compared with w_5. $\gamma_{i,j}(1)$ is ultimately computed after w_1 is compared with w_4, w_3, and w_2, which in our case is 46. Similarly, $\gamma_{i,j}(2)$, $\gamma_{i,j}(3)$, $\gamma_{i,j}(4)$, $\gamma_{i,j}(5)$, $\gamma_{i,j}(6)$ are computed by iterated the above procedures, which are 29, 10, 4, 1, and 0. Therefore, the importance level list $\{49, 36, 29, 10, 4, 1, 0\}$ is filled into a new list $f_{i,j}$, which is the fingerprint created for $a_{i,j}$.

The RSS correlation-based fingerprint is essentially an extension of the mobility signature introduced in Section 3.2.1. Both fingerprints and mobility signatures extract spatial information from raw RSS measurements by exploiting relative RSS correlations among Wi-Fi APs rather than absolute RSS values. As a result, invariant features in RSS data can be captured, mitigating the impact of RSS fluctuations caused by signal multipath propagation. Moreover, when information from a subset of Wi-Fi APs becomes unavailable due to dynamic radio environments, correlations among the remaining APs can still be leveraged for compensation. The key difference between fingerprints and mobility signatures lies in their correlation comparison mechanisms. Mobility signatures analyze RSS correlations using a predefined fixed

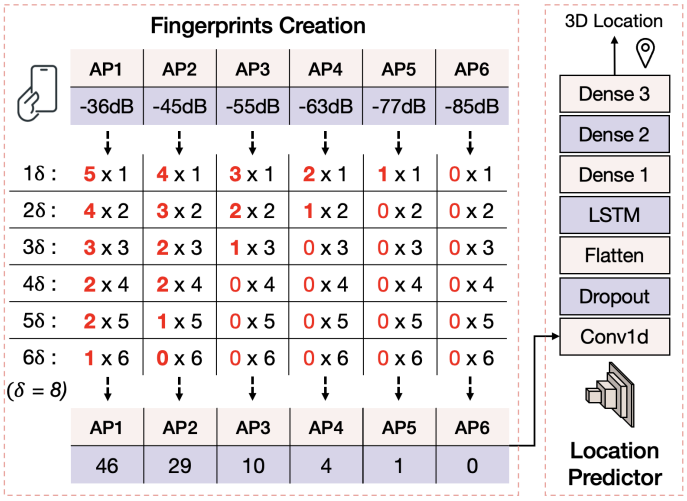


Figure 5.3: Creation of fingerprints.

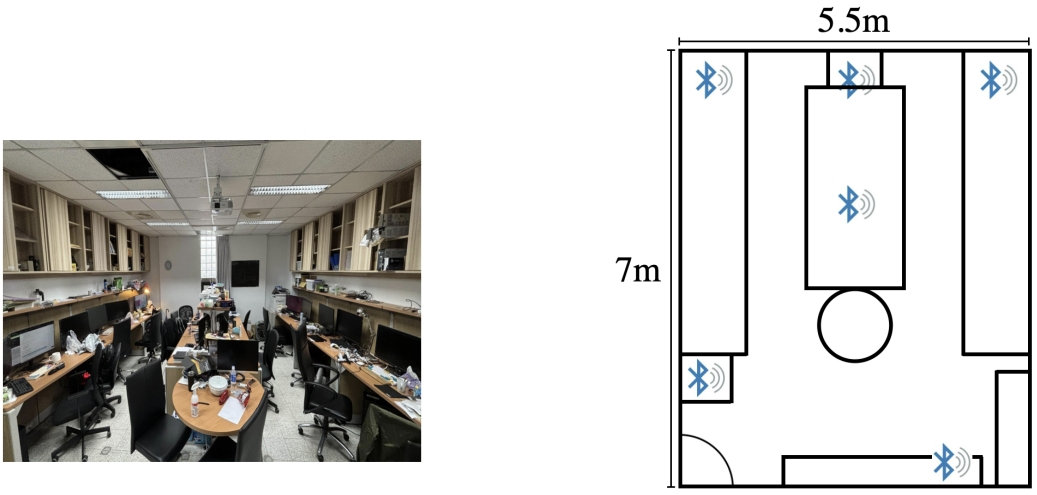
threshold, which is sufficient for capturing overall movement patterns and trajectory similarity. In contrast, fingerprints investigate finer-grained RSS correlations among APs using multiple dynamic thresholds implemented through a recursive function, thereby preserving richer spatial signal characteristics for accurate localization.

5.3.2 Location Predictor

This work designs a location predictor based on the created fingerprints to correlate the fingerprints with their corresponding locations, thus enabling accurate localization. As shown in Fig. 5.3, the devised location predictor consists of a convolutional layer, an LSTM layer, and three fully connected layers. While the fingerprinting technique extracts the local features from observed APs by checking their pairwise RSS correlation, the LSTM layer employed in the location predictor captures the global features from APs. In addition, the convolutional layer in the location predictor is utilized to help the LSTM better capture the short-term temporal dependencies. The fully connected layers are used to bridge the LSTM-predicted sequence and the fixed-size outputs. The dropout layer is used to prevent overfitting during training.

5.4 Experiments and Evaluation

Extensive experiments are conducted in this work based on three datasets of varying scales, where a small-scale dataset is collected in our Lab at the university, while the medium- and large-scale datasets are publicly open-sourced datasets. As demonstrated by the comprehensive experimental results in this section, the proposed system provides a substantially temporal-resilient system for wireless signal-based localization, outperforming SOTA methods. When the ambient environment changes over time, i.e., the disappearing of existing APs or the joining of new APs, our system



(a) University Lab. **(b)** Floor plan of Lab.

Figure 5.4: Small-scale environment.

effectively mitigates performance degradation by merely leveraging a few sparse observations from newly detected APs. In this section, we first comparatively study the overall performance of the proposed system and SOTA in a complex environment and over a long period. Next, we dive into the details and further investigate how our system alleviates the performance decay over time. Last, advanced experiments are made to evaluate the system's performance from different perspectives.

5.4.1 Experimental Setup

Dataset Descriptions

In this work, three datasets of varying scales are leveraged for a comprehensive evaluation of the performance of the proposed system, i.e., a large-scale dataset, a medium-scale dataset, and a small-scale dataset.

Large-Scale Dataset. The large-scale dataset [MRT+18] is created in a university library, where Wi-Fi samples are collected in a two-story area of 2 × 308.4 m^2 filled with bookshelves and passing people. The dataset is collected over 15 months in this area, during which 63,504 Wi-Fi samples are recorded and a total of 174 Wi-Fi APs are detected. We select 13 months of data from this dataset, i.e., from the previous January to the next January, and each month contains 3,120 Wi-Fi samples collected at 106 RPs as training data and 512 Wi-Fi samples collected at 24 RPs as test data.

Medium-Scale Dataset. The medium-scale dataset [HYD+19] is collected in an indoor office area of 16 m × 21 m. In this dataset, 365 RPs are uniformly selected from this office area to collect Wi-Fi samples for training, and another 175 RPs are chosen to collect Wi-Fi samples for testing. At each RP, multiple Wi-Fi

samples are effortlessly captured by a mobile phone mounted on an autonomous driving robot. This robot, equipped with different types of sensors, also records the two-dimensional coordinates of each Wi-Fi sample with a localization error of 0.07 m. In total, this dataset records 30,335 Wi-Fi samples for training and 3,120 Wi-Fi samples for testing, in which a total of 15 Wi-Fi APs are detected.

Small-Scale Dataset. As shown in Fig. 5.4, the small-scale dataset is created by ourselves in a Lab at our university, where 6 BLE beacons are deployed in the 5.5 m × 7 m Lab to verify that our system is also compatible with BLE signals. In this environment, we evenly select 94 RPs and record 32 BLE samples at each RP, with a total of 3,008 Wi-Fi samples collected by a smartphone.

Performance Metrics

This section introduces the four experimental metrics utilized for performance evaluation: network change level, localization error, performance decay, and RSS deviation.

- *Network change level.* During the collection of Wi-Fi samples in the first month, the set of all detected Wi-Fi APs is denoted by S_1. Similarly, we denote by S_i the set of all APs detected during the collection of Wi-Fi samples in the i-th month. (Both S_1 and S_i only record the unique BSSID of each detected AP). The network change level is defined as $1 - \frac{|S_1 \cup S_i|}{|S_1|}$, which quantifies the variation in detectable Wi-Fi APs from the first month to the i-th month.
- *Localization Error.* Localization error is a widely adopted metric in localization research, which describes the Euclidean distance between the true coordinates (ground truth) and the coordinates estimated by the localization system.
- *Performance Decay.* Let L_1 denote the localization error of a localization system in the first month and L_i denote the localization error in the i-th month. The performance decay in the i-th month is defined as $\text{Max}(0, L_i - L_1)$, representing the decay in system accuracy over time.
- *RSS Deviation.* We denote by $R_f = \{r_1, r_2, ..., r_N\}$ the RSS values generated by a generative model at N RPs and by $R_g = \{\hat{r}_1, \hat{r}_2, ..., \hat{r}_N\}$ the actual RSS values at these N RPs. The RSS deviation is defined as the absolute difference between R_f and R_g, which describes how far the generated RSS values deviate from the actual values.

5.4.2 Network Implementation

Fig. 5.2 and Fig. 5.3 introduce the overall network architecture of the devised radio map generator, radio map discriminator, and location predictor. This section further describes the implementation details of each network, e.g., the kernel size, the channel size, and the length of strides, etc.

Implementation of Generator

Fig. 5.2 presents the network structure of the radio map generator. The input of the generator is a tensor of dimension (64, 1, 64, 64) representing that for each input, the batch size is 64, the channel size is 1, and the image size is 64×64. Table 5.1 presents all the parameters of the 11 convolution layers used in the generator. As can be seen in the U-shaped generator, the channel size first increases from 1 to 128 and then decreases from 128 to 1, while the image size decreases from 64×64 to 16×16 and then increases from 16×16 to 64×64. The discriminator and generator are trained collaboratively, with the generator trained only once after the discriminator is trained five times. During the training of the generator, the learning rate is set to 0.0001, the penalty coefficient for the gradient penalty loss is set to 10, and the number of training epochs is 150.

Table 5.1: Network architecture of the generator.

Conv2d	Channel size	Kernel size	Padding	Stride	Image size
1	32	3	1	1	64×64
2	32	3	1	1	64×64
3	64	3	1	2	32×32
4	64	3	1	1	32×32
5	128	3	1	2	16×16
6	128	3	1	1	16×16
7	64	3	1	1	32×32
8	64	3	1	1	32×32
9	32	3	1	1	64×64
10	32	3	1	1	64×64
11	1	1	0	1	64×64

Implementation of Discriminator

Fig. 5.2 also shows the network structure of the radio map discriminator. The input of the discriminator has one more channel than the generator because it takes both the image generated by the generator and the real image as input. Therefore, the input of the discriminator is a tensor with a dimension of (64, 2, 64, 64). Table 5.2 introduces the parameters of the convolution layers in the discriminator. As can be seen, the channel size keeps decreasing while the image size increases until an output of size 1×1 is produced, which quantifies the divergence between the generated image and the real image. The discriminator shares the same batch size, learning rate, gradient penalty coefficient, and number of epochs as the generator. Note that the discriminator employs multiple dropout layers to avoid overfitting during training, with a dropout rate of 0.5.

Table 5.2: Network architecture of the discriminator.

Conv2d	In/Out channel	Kernel size	Padding	Stride	Image size
1	16	4	1	2	32×32
2	32	4	1	2	16×16
3	64	4	1	2	8×8
4	128	4	1	2	4×4
5	1	4	0	1	1×1

Implementation of Location Predictor

Fig. 5.3 demonstrates the network architecture of the location predictor. The 1D convolution layer has a channel size of 8, a kernel size of 1, a stride of 1, a padding of 0, and an activation function of eLU. The LSTM layer has a channel size of 50, an activation function of tanh, and a recurrent activation function of sigmoid. The channel sizes of the three dense layers are 16, 10, and 3, respectively, and their activation functions are all eLU. During the training of the location predictor, the learning rate is set to 0.005, the number of epochs is 200, and the batch size is 256.

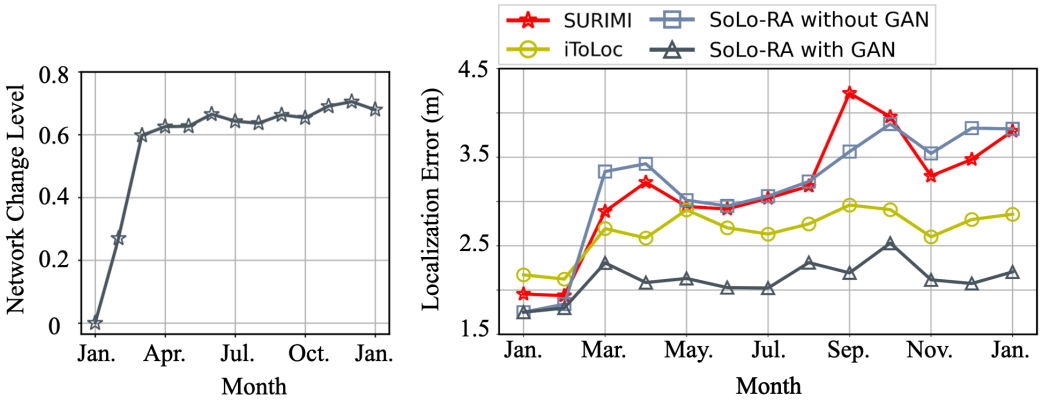
5.4.3 Comprehensive Analysis in a Large-Scale Environment

In this section, we evaluate the localization performance of our system from various perspectives using the large-scale dataset. First, we observe the changes in detectable Wi-Fi APs in the environment over a period of more than a year. Next, we comparatively analyze the impact of such changes on the localization performance of our system and SOTA. To adapt to the changes in WiFi APs and thereby mitigate performance decay, our system trains a generative model to generate radio maps for newly observed APs. We subsequently explore the correlation between performance decay and radio map generation. Moreover, we study the correlation between performance decay and wireless network dynamics, i.e., the number of changed Wi-Fi APs. In the end, we investigate the impact of the number of Wi-Fi APs learned by the generative model.

Long-Term Analysis of Performance Evolution

In this section, we comparatively investigate the performance changes of localization systems over time. The comparative SOTA is Surimi [QTN+22] and iToLoc [LXY+24]. The experiments in this section are conducted based on the Wi-Fi data collected over a period of 13 months in the large-scale dataset, as described in more detail in Section 5.4.1. Initially, the Wi-Fi samples collected in the first month (January) are leveraged to train individual localization models for Surimi, iToLoc, and our system. Meanwhile, our system trains a generative model for radio map generation also based on the Wi-Fi samples of January. Subsequently,

the localization models developed in January are tested using the Wi-Fi samples collected over the next 12 months (February to next January), for the purpose of analyzing the performance changes over time. Note that when new APs are detected during the test stage, our system employs the trained generative model to generate dense radio maps for them.



(a) Network change level over a year.

(b) Changes in localization error.

Figure 5.5: Overall performance comparison with SOTA over a year.

The experimental results are shown in Fig. 5.5(b), where the curves with circles and asterisks represent the performance of iToLoc and Surimi, respectively. The curves with squares and triangles show the performance of our system before and after the generation of radio maps for the newly detected APs. As can be seen, Surimi, iToLoc, and our system all achieve satisfactory localization accuracy in January in the complex multi-floor 3D environment, with an average localization error of 1.75 m for our system, 1.955 m for Surimi, and 2.172 m for iToLoc. In February, the three localization models still keep lower localization errors, with 1.841 m for our system, 1.937 m for Surimi, and 2.124 m for iToLoc. It is worth noting that in March, the network change level increases from 0.269 to 0.597, as shown in Fig. 5.5(a). This indicates that the APs detected in March differ greatly from those detected in January or February. Consequently, the system performance is significantly hampered when the localization model trained in January is employed to locate the Wi-Fi samples collected in March. As expected, the average localization errors of our system, Surimi, and iToLoc respectively increase to 3.338 m, 2.888 m, and 2.694 m in March, as shown in Fig. 5.5(b). To compensate for the performance decay, our system generates radio maps for the newly detected APs and then retrains our localization model by incorporating the generated radio maps. As a result, the localization error of our system is greatly reduced from 3.338 m to 2.307 m in March. From April to next January, the average localization error of our system is also reduced from 3.430 m to 2.170 m, which is significantly lower than the 3.402 m of Surimi and the 2.770 m of iToLoc over the same period.

To summarize, Surimi performs the worst, and iToLoc, while outperforming Surimi, is still much inferior to our system. The above results can be explained with the help of the two potential reasons causing the performance decay, as we discussed in Chapter 1. First, the RSS of an AP may vary from time to time, even at the same location. This is because objects or people in the environment could be at different locations at different times, thus causing different signal fading. Second, the detectable APs in the same environment may also differ from time to time. iToLoc can promptly and effortlessly update the RSS of APs through semi-supervised learning, thus alleviating the performance decay caused by signal fading. However, the semi-supervised learning in iToLoc is incapable of learning newly detected APs. Hence, the system performance is still impaired when the network change level is high. Surimi merely leverages the GAN to generate denser Wi-Fi FPs effortlessly. Nevertheless, when the Wi-Fi samples utilized to develop the localization model are already dense enough, Surimi can no longer generate more useful FPs. Therefore, Surimi can address neither of the two aforementioned problems.

Correlation Study between Performance Decay and APs Generation

In the previous section, our system is validated to be temporally robust against the changes of APs in the environment, which is achieved by generating radio maps for the newly observed APs. In this section, we expound further on how our system progressively alleviates performance decay via the generation of radio maps, i.e., to investigate the correlation between performance decay and the number of generated radio maps. First, we study in depth the correlation between the number of generated APs and performance decay. Then, we analyze the correlation between the number of radio maps learned by our system and the quality of the generated radio maps.

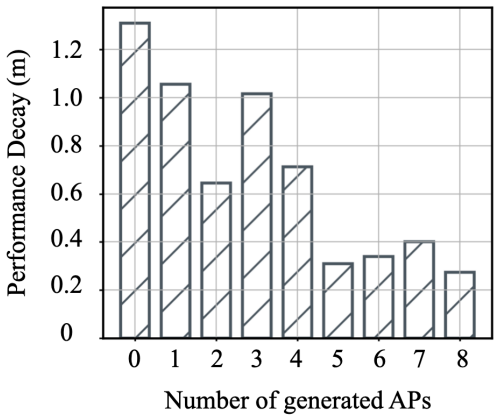


Figure 5.6: Performance decay vs. number of generated APs.

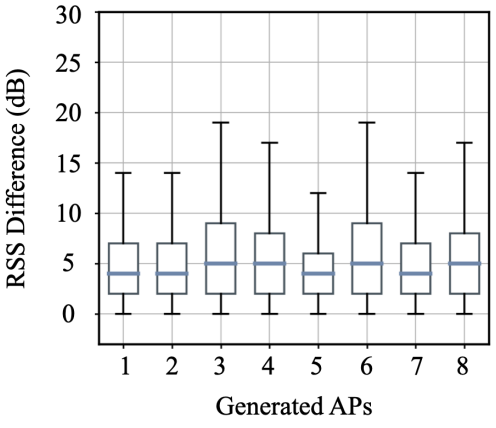


Figure 5.7: Quality analysis of generated APs.

To gain a better insight into the impact of generating radio maps for new APs on mitigating performance decay, we first observe the APs detected in any two months, e.g., January and July. As observed, 8 new APs are detected in July compared to January. As a consequence, when we employ the localization model trained in January to locate the Wi-Fi samples collected in July, and no radio maps are generated for any new APs, the performance decay sharply increases to 1.310 m, as shown in Fig. 5.6. Fortunately, our system is capable of mitigating performance decay by generating radio maps. When we generate the radio map only for 1 out of the 8 APs and retrain the localization model by incorporating the generated radio map, the performance decay is reduced from 1.310 m to 1.055 m. When we generate radio maps for 2, 3, 4, 5, 6, 7, and 8 APs, respectively, the performance decay is accordingly reduced to 0.645 m, 1.017 m, 0.713 m, 0.31 m, 0.34 m, 0.401 m, and 0.273 m. As can be seen, the performance decay of our system is getting smaller as the number of generated radio maps increases. The performance decay reaches a minimum of only 0.273 m when radio maps are generated for all 8 APs.

The mitigation of performance decay of our system is highly correlated with the quality of the generated radio maps. To analyze the quality of the generated radio maps, we compute the RSS deviation of each generated radio map from its ground truth. As shown in Fig. 5.7, the average RSS deviation of radio maps generated for all APs is smaller than 5 dB, where RSS deviation for AP1, AP2, AP5, and AP7 is even smaller than 4 dB. Note that in an indoor environment, two consecutive RSS samples taken at even the same location from a certain AP may be greater than 5 dB. The fingerprinting technique proposed in this work is resistant to RSS with deviations within 5 dB, therefore our system achieves a promising performance aided by the generated radio maps.

Impact of Wireless Network Dynamics

This work refers to wireless network dynamics as changes in detectable Wi-Fi APs in the environment. In this section, we further explore the correlation between our system's performance decay and the number of disappeared APs. First, while keeping the test set complete and fixed, we randomly remove 8 APs from the training set and then train a localization model and a generative model for radio map generation. Note that both the training set and test set are the Wi-Fi data collected in January from the large-scale dataset in this section. Next, we retrieve sparse RSS samples of these 8 APs from the complete and fixed test set, which are then utilized by the generative model to generate denser radio maps for these 8 APs. We update the training set by replacing the 8 removed APs with generated radio maps. Subsequently, the updated training set is leveraged to retrain the localization model. In the end, we compare the localization performance after removing APs and the performance after generating radio maps. We execute the above steps repeatedly while removing 1, 2, 3, 4, 5, 6, and 7 APs, respectively, so as to compare the system performance after removing 1 to 7 APs and the performance after generating 1 to 7

radio maps. To rule out experimental flukes, the above procedures are repeated 12 times, using the Wi-Fi data from February to the next January, respectively.

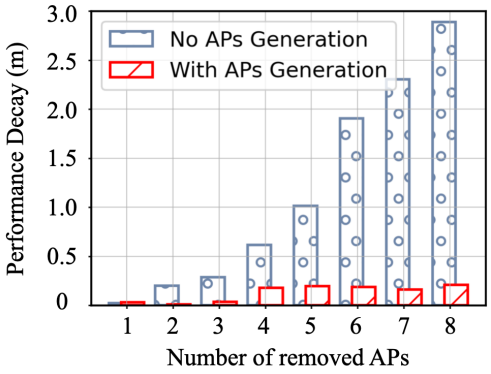


Figure 5.8: Performance decay vs. number of generated APs.

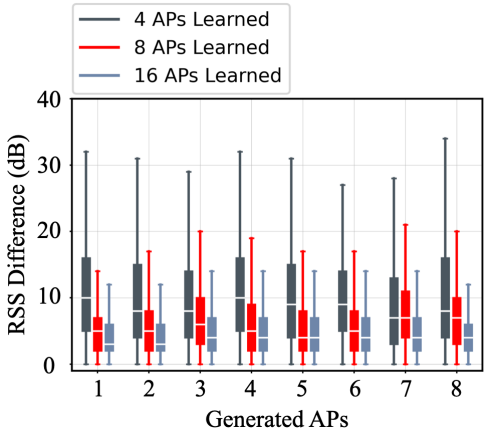


Figure 5.9: Quality analysis of generated APs.

As demonstrated in Fig. 5.8, the performance decay of our system is 0.022 m, 0.201 m, 0.287 m, 0.615 m, 1.014 m, 1.904 m, 2.305 m, and 2.887 m when the number of disappeared APs ranges from 1 to 8, respectively. As can be seen, the system performance is barely impaired when only one AP disappears, with a performance decay of 0.022 m. In this situation, generating the radio map for the AP instead increases the performance decay to 0.029 m. When the number of disappeared APs increases to two or three, the system performance is slightly hampered, with performance decay of 0.201 m and 0.287 m, respectively. It is then visibly observed that generating radio maps for disappeared APs reduces the performance decay from 0.201 m to 0.01 m, and from 0.287 m to 0.034 m. The system performance deteriorates increasingly as more APs disappear. When the number of disappeared APs is 4, 5, 6, 7, and 8, the performance decay severely increases to 0.615 m, 1.014 m, 1.904 m, 2.305 m, and 2.887 m. Nevertheless, benefiting from the generation of radio maps, our system can alleviate the performance decay to 0.176 m, 0.193 m, 0.186 m, 0.162 m, and 0.207 m, respectively.

In conclusion, regardless of the number of disappeared APs, our system can greatly redeem the system performance through the generation of radio maps. Particularly when more APs disappear over time, the generation of radio maps is imperative to prevent the localization system from crashing.

Impact of Number of learned APs

As we explained in Chapter 1, the fundamental rationale behind the devised generative model is to learn the propagation behavior of Wi-Fi APs in the target environment. We assume that the more APs in this environment are learned, the

closer the generated radio maps resemble the actual radio maps. To validate our assumptions, we investigate the correlation between the quality of the generated radio maps and the number of learned APs in this section. The experimental results are based on the large-scale dataset consisting of 174 detectable APs. Note that we find through observations that not all APs can serve as dense radio maps to be learned by the generative model. First, many APs are inconsistently detectable throughout the Wi-Fi data collection stage and only produce highly sparse Wi-Fi samples. Second, the RSS of numerous detected APs varies within a narrow range, often below -70 dBm. This occurs because Wi-Fi APs typically provide coverage over a large area of about 100 m, and many detected APs are deployed outside the target environment, resulting in smaller RSS values. As a result, only approximately 20 APs in the large-scale dataset can be utilized to train our generative model.

Fig. 5.9 demonstrates the RSS deviation of the generated radio maps compared to the real radio maps, when the generative model learns 4 APs, 8 APs, and 16 APs, respectively. As can be seen, the quality of the generated radio map is unsatisfactory when the generative model only learns 4 APs. In comparison to ground truth, the average RSS deviation is about 10 dB, and the maximum RSS deviation even exceeds 30 dB. This is because the number of learned APs is too small, and the generative model is incapable of comprehensively learning the signal propagation in each area of the target area. As the number of learned APs increases to 8, the average RSS deviation is reduced to about 5 dB, and the maximum RSS deviation is smaller than 20 dB. By this time, APs are distributed in many areas of the target environment as the number of APs increases, and the generative model can roughly grasp the signal propagation behavior in the target environment. Eventually, when the number of learned APs is increased to 16, the average and maximum RSS deviation are reduced to 3 dB and 12 dB, respectively. This indicates that the generative model learns sufficient APs to even characterize the signal propagation details in the target environment.

5.4.4 Advanced Study in various Environments

In this section, we evaluate the average localization error of our system compared to SOTA using different datasets in different environments. Consequently, we validate that our system is not only temporally but also spatially robust.

Localization Error in various Environments

Table 5.3 shows the average localization errors of our system and SOTA in the small-scale environment, medium-scale environment, and large-scale environment, respectively. Fig. 5.10 further demonstrates the CDF of the localization errors of our system and SOTA in these different environments. As shown in Table 5.3, the average localization error in the small-scale environment is 0.773 m for our system, 0.834 m for Surimi, and 0.809 m for iToLoc. In the medium-scale environment, the average localization error is 1.488 m for our system, 1.695 m for Surimi, and

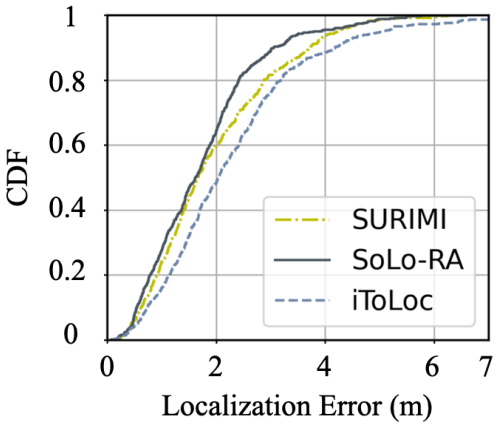
Table 5.3: Average localization error (m).

	SoLo-RA	**Surimi**	**iToLoc**
Small-scale dataset (BLE)	0.773	0.834	0.809
Medium-scale dataset (Wi-Fi)	1.488	1.695	1.717
Large-scale dataset (Wi-Fi)	1.750	1.955	2.167

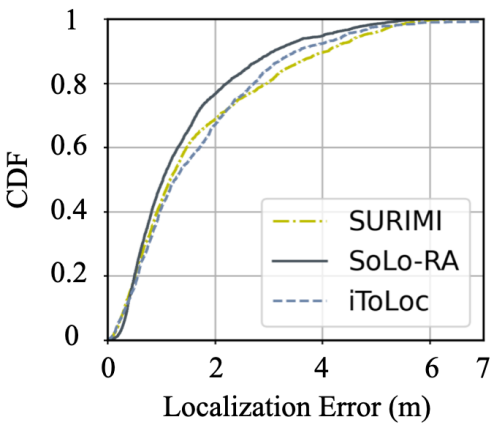
1.717 m for iToLoc. In the large-scale environment, the average localization error is 1.750 m for our system, 1.955 m for Surimi, and 2.167 m for iToLoc. As can be observed, our system significantly outperforms Surimi and iToLoc in all environments. Fig. 5.10 further verifies that our system is superior to SOTA. As shown in Fig. 5.10a, Fig. 5.10b and Fig. 5.10c, the CDF curves of our system are consistently higher than that of Surimi and iToLoc in all environments. In the small- and medium-scale environments, the performance of Surimi is slightly better than iToLoc because the curves representing Surimi converge earlier than the curves of iToLoc. In the large-scale environment, the performance of Surimi is always superior to iToLoc. Consequently, similar results demonstrate that our system has the best performance, followed by Surimi, and iToLoc performs the worst.

Our system outperforms Surimi primarily on account of the devised fingerprinting technique. Surimi directly leverages the raw RSS data to train the localization model. However, the detected RSS may fluctuate greatly even at the same location due to signal fading in the environment, which degrades the performance of the trained localization model. In contrast, our fingerprinting technique explores the RSS correlation between different APs rather than the absolute RSS value of each AP. Therefore, although an AP's RSS constantly fluctuates due to signal fading, our system can still reliably assess the RSS correlation between this AP and other APs, as long as the fluctuation is within the range of δ.

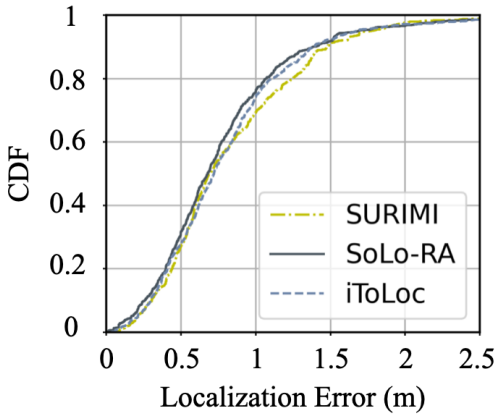
The poorest performance of iToLoc complies with our expectations. While iToLoc encodes the RSS correlation between different APs into an image for fingerprint construction, it also encodes large amounts of irrelevant background information. To be specific, a total of 174 APs are detected in the large-scale data set, and iToLoc accordingly creates a fingerprint image with a size of 174×174. In this image, the useful information, consisting of the RSS correlation between APs, is extremely scarce compared to the vast amount of pixel-0 background information. Moreover, while creating dense fingerprint datasets for the environment, 174×174 images are computationally expensive for both fingerprint generation and model training.



(a) Localization accuracy in a large-scale environment.



(b) Localization accuracy in a medium-scale environment.



(c) Localization accuracy in a small-scale environment.

Figure 5.10: Localization accuracy in different environments.

Analysis on Fingerprint Generation

In the previous sections, newly observed Wi-Fi APs are taken into account. Our system generates radio maps for the newly observed APs and then updates the fingerprint dataset with the generated radio maps. In this section, the changes in Wi-Fi APs are not considered, and our system directly generates a denser fingerprint dataset for the target environment. Then, we investigate the performance improvement associated with the number of generated fingerprints.

In this section, we again utilize the large-scale dataset to assess the impact of generated fingerprints on system performance. Notably, to provide sufficient ground truth for evaluation, we take the original test set as the training set, which includes 512 Wi-Fi samples collected at 24 RPs. Similarly, we take the original training set as the test set, which contains 3120 Wi-Fi samples collected at 3120 RPs. First, a

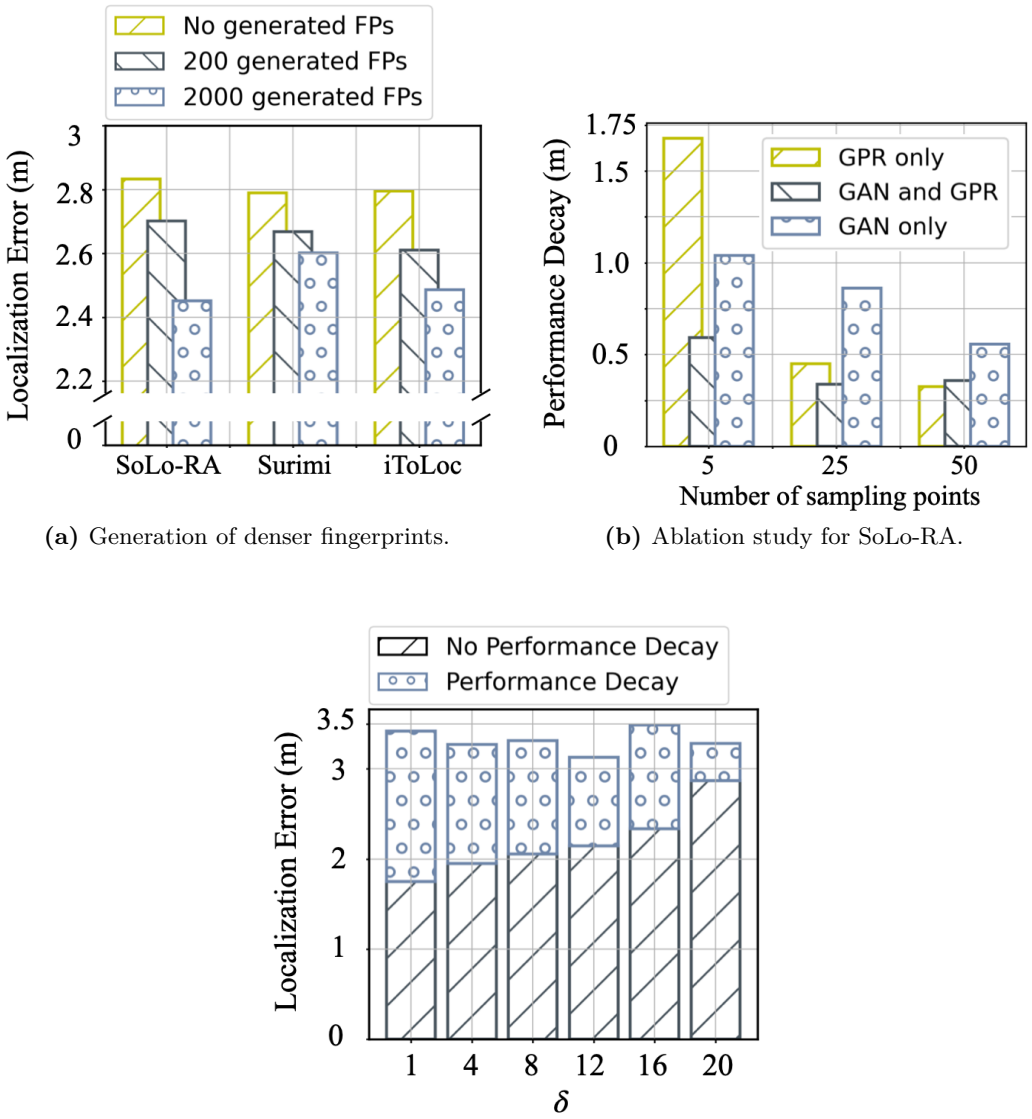
localization model and a generative model are trained for our system, Surimi, and iToLoc, respectively, using the 512 Wi-Fi samples collected at 24 RPs. Second, the three trained generative models are employed to generate denser fingerprints. Last, the generated fingerprints are incorporated into the training set, and the localization models are retrained.

As shown in Fig. 5.11a, while no additional fingerprints are generated, our system's localization error is 2.832 m, which is slightly higher than Surimi's 2.790 m and iToLoc's 2.796 m. When 200 fingerprints are generated, the localization error of our system decreases significantly to 2.702 m. However, it still performs the worst, as the localization errors of Surimi and iToLoc further decrease to 2.668 m and 2.611 m, respectively. Notably, when 2000 fingerprints are generated, our system's localization error drops substantially to 2.452 m, while the errors of Surimi and iToLoc only show slight reductions to 2.602 m and 2.486 m. Such results are in line with our assumptions. As the training set for the generative model only contains the Wi-Fi samples collected at 24 RPs, the generation target is the Wi-Fi samples at 82 RPs outside the training set. Surimi and iToLoc, which employ GANs or other classification models, can hardly generate correct Wi-Fi samples beyond the training distribution. iToLoc is slightly better than Surimi because it uses semi-supervised learning, which empirically enriches the model with extra knowledge. In comparison to SOTA, our generative model synergistically utilizes GAN and GPR, which can not only capture high-dimensional features in the training set, but also compensate for the limited extrapolation ability of the generative model aided by the regression technique.

Ablation Study

In this section, we conduct ablation experiments while varying the number of RPs, at which Wi-Fi samples are collected for newly observed APs. On one hand, we investigate the correlation between system performance and the number of RPs, because the quality of generated radio maps is highly related to the sparseness of Wi-Fi samples. On the other hand, we study the individual importance of GAN and GPR from the perspective of RP sparseness, as they exhibit distinct characteristics under different RP sparseness.

Fig. 5.11b presents the experimental results. With Wi-Fi samples collected at only 5 RPs, the localization error of our system reaches up to 3.428 m when using only GRP to generate radio maps. In comparison, a lower localization error of 2.789 m is observed when employing GAN alone. Notably, When GPR and GAN are utilized collaboratively, the localization error is lowest at 2.342 m. This is because regression techniques are heavily dependent on observation data to deduce functions. Specifically, GPR uses kernel functions to capture the correlation between 2D coordinates in this work. When the data is extremely sparse, the ability of kernel functions to characterize the spatial correlation between 2D coordinates is limited. As a result, GPR exhibits severely limited generation capability when the



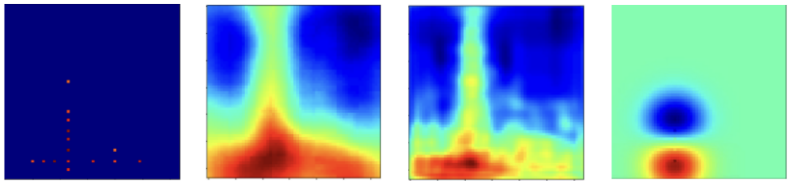
(a) Generation of denser fingerprints.

(b) Ablation study for SoLo-RA.

(c) Impact of δ selection.

Figure 5.11: Advanced study in different environments.

observation data is highly sparse. However, as the generative model learns how signals propagate in the target environment by utilizing extensive training data during the offline phase, it can create dense radio maps for new APs despite their sparse observation data. Fig. 5.12 gives an example of radio map generation when RPs are very sparse. As observed, the radio map generated by our system, i.e., the collaboration of GAN and GPR, closely resembles the actual radio map. Conversely, the radio map generated by GPR deviates significantly from the actual radio. As shown in Fig. 5.11b, the localization errors decrease accordingly as the number of



(a) Sparse Wi-Fi samples. (b) Actual radio map. (c) Radio map generated by SoLo-RA. (d) Radio map generated by GPR.

Figure 5.12: An example of generation with sparse Wi-Fi samples.

RPs increases. It is worth noting that when only using GRP for generation, the localization error is dramatically reduced to 2.076 m meters, which is lower than the 2.109 m of our system. This is because GPR is capable of learning the spatial relationship between 2D coordinates when observations are sufficiently dense, leading to smoother interpolation between data points. The generative model in our system takes the radio maps generated by GPR with dense observations as the learning target.

Analysis on Parameter Selection

Toward the close of the experimental section, we introduce a pivotal feature of the devised fingerprinting technique, that is, the trade-off between localization accuracy and temporal robustness can be achieved by tuning δ. Fig. 5.11c presents the system performance of our system under different δ. As can be seen, the lower part of the histogram in Fig. 5.11c represents the average localization error of our system in the large-scale environment in January. The upper part of the histogram shows the localization error for the next 12 months. As δ increases from 1 to 20, the localization error of our system progressively increases from 1.750 m to 2.869 m. On the contrary, the performance decay of our system gradually drops from 1.672 m to 0.412 m as δ increases. The minimum localization error of 1.750 m is observed when δ is set to 1. However, the performance decay caused by signal fading is maximal in this case. In comparison, when δ is set to 20, although the performance decay is a minimum of 0.412 m, the localization error reaches a maximum of 2.869 m.

The experimental results comply with the underlying principle of the fingerprinting technique devised in our system. When a large δ is selected, our method tolerates the RSS oscillation within the range $[-\delta, \delta]$, at the expense of the accuracy of the localization model. This is because a larger δ increases the uncertainty of different classes in our localization model, thus weakening the prediction ability of the model. While localization performance improves when we opt for a smaller δ, the tolerance for RSS fluctuations diminishes, causing severe impairment to the localization model caused by signal fading. Therefore, we need to strike a trade-off between localization accuracy and temporal robustness. As shown in Fig. 5.11c, the trade-off is observed when δ is around 12, where the localization error is 2.145 m,

and the system degradation is 0.983 m. In this case, our system exhibits temporal robustness while achieving an acceptable localization error.

5.5 Conclusion

This work develops a spatial reconstruction and localization framework based on radio map adaptation. First, a radio map generator is trained in this framework through the joint collaboration of a GPR model and a GAN-based model. The generator learns the sparse-to-dense evolution of APs' radio maps, and can therefore generate full radio maps for APs using their sparse RSS observations. Second, a novel fingerprinting algorithm is additionally devised in the framework, by investigating the relative RSS correlation between different APs. Extensive experimental results demonstrate that the proposed framework provides a spatiotemporal robust scheme for indoor Wi-Fi localization.

Chapter 6

Conclusions, Discussion, and Outlook

6.1 Summary

This dissertation proposes a collaborative and complementary computing paradigm for mobility analytics based on passive sensing data. Based on the proposed computing paradigm, this dissertation performs mobility analytics from three different perspectives, i.e., complementation between multi-modal sensing data, collaboration between cross-domain sensing data, and multi-model-based generation of sensing data.

Mobility Trajectory Comparison Based on Multi-Modal Sensing Data

This dissertation explores human proximity using multi-modal sensing data from different sensors, i.e., Wi-Fi data from ambient Wi-Fi APs and acceleration data from the user's built-in sensor. This work models the collected Wi-Fi data into mobility signatures and provides users with macroscopic mobility information. Also, this work models the collected IMU information into movement signatures and provides users with microscopic movement information. Mobility information implies users' spatial relationships, but it lacks the ability to perceive small movements. In contrast, movement information is sensitive to subtle body motions, but lacks the support of relative or absolute location. Therefore, this dissertation, on the one hand, jointly utilizes users' mobility signatures and movement signatures to infer their social distance, where the Wi-Fi data and the IMU data complement each other for their respective weaknesses. On the other hand, this dissertation considers consecutive mobility signatures and consecutive movement signatures as users' mobility trajectories. By quantifying the similarity between users' mobility trajectories, this dissertation builds a bridge between the cyber-world distance and physical-world distance. Extensive experimental results not only demonstrate the importance of utilizing multi-modal sensing data, which improves the accuracy of proximity detection, but also validate the ability of the proposed metrics (inter-similarity, self-similarity, and movement similarity) in establishing a linkage between cyber-world and physical-world.

Visual Trajectory Comparison Based on Cross-Domain Sensing Data

This dissertation investigates the correlation between users' visual attention through the collaboration between cross-domain passive sensing data, i.e., users' eye movements and the light patterns reflected in their eyes. In this work, an eye camera is employed to continuously capture information of the user's eyes. Although both the movement of the eyes and the light reflected in their eyes originate from the same sensor, their perception of the environment is characterized from different knowledge domains. The collaborative use of these two types of data in this dissertation can facilitate a more comprehensive detection of visual attention. Specifically, the goal of this dissertation is to use the captured information from users' eyes to determine whether users have the same visual attention, e.g., viewing the same painting or watching the same TV show. To this end, this dissertation models the light reflected in the user's eyes as continuous visual signatures, where each visual signature is the frequency domain distribution of the light. When the user's visual attention changes, the light reflected in his eyes also evolves, thus generating continuous visual signatures. This dissertation refers to the continuous visual signatures of the user as his visual attention trajectory. By quantifying the similarity between the visual attention trajectories of users, the correlation between their visual attention can be investigated. This dissertation verifies through extensive experiments that the proposed visual signatures can model the light patterns reflected in the user's eyes effectively, and the proposed similarity quantification algorithms can also reliably investigate the correlation between users' visual attention.

Multi-Model-Based Generation of Sensing Data

This dissertation reveals two important challenges in Wi-Fi-based indoor localization through extensive experiments: 1) offline collecting Wi-Fi fingerprints requires large amounts of human effort. 2) constructed Wi-Fi fingerprint database may deteriorate over time, because changes in wireless network conditions are inevitable. Therefore, this dissertation designs a framework for Wi-Fi fingerprint generation and indoor localization, based on AI-driven models and non-AI-driven models. AI models are usually better at extracting high-dimensional features from data and are therefore superior at generating the details of Wi-Fi fingerprints. Non-AI models possess strong extrapolation capabilities, and can still generate a reasonable result when the Wi-Fi fingerprint generation framework is fed with unlearned targets. Therefore, this dissertation collaboratively utilizes AI-driven models and non-AI-driven models to generate Wi-Fi fingerprints, thereby enhancing the robustness of Wi-Fi-based indoor localization. In addition, this dissertation designs a novel Wi-Fi-based indoor localization algorithm that can provide satisfactory localization results even without the generation of fingerprints. Finally, this dissertation also verifies through extensive experiments that the performance of the designed fingerprint-based localization system is not significantly degraded compared to existing state-of-the-art research and can stand the test of time.

6.2 Discussion

This section first discusses the extensibility of the proposed mobility analytics framework, with a focus on the feasibility of incorporating both multi-modal and cross-domain sensing data. It then investigates how the proposed framework can be generalized to additional sensing modalities, such as acoustic signals. Finally, it discusses the limitations of deep learning-based methods in this dissertation.

Incorporating Multi-Modal and Cross-Domain Sensing Data

In this dissertation, mobility analytics is performed based on sensing data collaboration and complementation for two reasons. First, the availability of sensing data is generally uncontrollable, because passive sensors in existing environments are deployed for purposes other than mobility analytics and may therefore become unavailable at any time due to removal, malfunction, or reconfiguration. Second, compared with active sensing-based mobility analytics, where dedicated sensors are deployed for specific applications, a single sensing modality or sensing domain in passive sensing-based mobility analytics often provides only limited information about user mobility. Therefore, this dissertation investigates how multi-modal sensing data or cross-domain sensing data can be leveraged, under constrained sensing conditions, to compensate for the absence of single-modality or single-domain sensing data and to mitigate potential system degradation.

As future work, jointly incorporating both multi-modal sensing data and cross-domain sensing data could further enhance the performance of mobility analytics when the availability of multiple sensor types or multiple sensing modalities can be reliably ensured. However, in practice, it is difficult to guarantee the continuous availability of both multi-modal sensing data and cross-domain sensing data due to the aforementioned two reasons. In addition, incorporating both multi-modal and cross-domain sensing data may incur higher deployment, processing, and maintenance costs, which conflict with the key advantage of passive sensing-based mobility analytics, namely its low overhead and cost effectiveness.

Toward General Mobility Analytics across Sensing Modalities and Domains

Although the proposed methods for mobility trajectory comparison and visual trajectory comparison are applied to different sensing modalities, namely Wi-Fi and visual data, their key methodological principles are highly consistent. In both cases, the raw sensing data are first transformed into structured representations, referred to as mobility signatures and visual signatures, which compactly capture users' mobility-related characteristics while filtering out irrelevant information. These signatures serve as modality-specific representations that unify heterogeneous sensing data into a common analytical form. Subsequently, temporally consecutive signatures

are organized into trajectories, and DTW-based algorithms are employed to match asynchronous trajectories collected at different sampling rates or time offsets.

Based on this shared two-stage modeling and matching approach, the proposed framework can be extended to additional sensing modalities or sensing domains. As future work, for any new modality, the raw sensing data can first be modeled into a corresponding signature that encodes mobility-relevant information. Once converted into a sequence of signatures, the resulting trajectory can be analyzed using the same DTW-based trajectory matching algorithm without changing the key comparison logic. For example, acoustic sensing data can be transformed into acoustic signatures based on sound intensity distributions or spectral features associated with user movement, and these signatures can then be matched over time using DTW to infer mobility similarity.

Limitations of Deep Learning-Based Methods in Passive Mobility Analytics

Recently, deep learning-based methods have demonstrated strong capabilities in modeling complex and nonlinear features and have been successfully applied to a wide range of mobility analytics applications. In principle, well-trained deep learning models could achieve higher performance when sufficient, well-labeled, and consistently available training data are provided.

However, in the context of this dissertation, deep learning-based methods face several practical limitations. First, they typically require large amounts of labeled training data, which is difficult to obtain for trajectory-based mobility analytics. Since the problem is to compare users' mobility trajectories rather than isolated data samples, the number of possible trajectory combinations grows combinatorially with trajectory length and user population. Collecting ground-truth similarity labels for such a vast number of trajectory pairs would be extremely labor-intensive and infeasible, which limits the availability of sufficient training data for effective model learning. Second, even if a deep learning model is trained through extensive data collection efforts, its performance may degrade significantly when the sensing environment changes. Models trained in one environment often do not generalize well to others due to differences in infrastructure, spatial layouts, and sensing conditions, which require repeated data collection and retraining, leading to additional efforts. Furthermore, training deep learning models generally incurs significant computational overhead, particularly in large-scale environments with extensive training data.

In contrast, the proposed methods are plug-and-play and operate without offline training, significantly reducing computational overhead. Although deep learning-based approaches may achieve higher performance under ideal conditions with sufficient data and stable environments, the cost of data collection, retraining, and computation may outweigh their potential benefits. Therefore, deep learning-based methods are better considered as a complementary direction rather than a replacement for the proposed framework, for example by being integrated into

specific stages to learn more expressive mobility-specific representations from raw sensing data.

6.3 Future Work

Mobility analytics based on wireless signals has been a central theme of this dissertation, with a particular focus on exploiting RSS measurements from ambient Wi-Fi APs to achieve efficient and cost-effective mobility analytics. Despite its practicality, Wi-Fi-based mobility analytics still has several limitations. Most notably, its performance depends strongly on the density of Wi-Fi APs in the environment. When a sufficient number of APs is available, the uncertainty in the received Wi-Fi measurements can be effectively reduced, leading to more reliable analytics results. In contrast, in environments with sparse Wi-Fi AP deployment, or even in extreme cases where only a single Wi-Fi AP is present, reliably estimating users' spatial locations becomes more difficult. To overcome these limitations, future work could investigate the use of additional sensors or complementary sensing modalities alongside Wi-Fi RSS, in order to improve the robustness and accuracy of mobility analytics.

Future Work on Multi-Modal Sensing Data

One promising direction is to leverage ambient light as a cost-effective and low-effort sensing modality [ZWZ+17; ZZ17]. Modern smart devices are commonly equipped with built-in light sensors, which can be readily used to collect light intensity measurements without additional hardware. Since light intensity generally decreases with increasing distance from the light source, such measurements can provide useful spatial information for mobility analytics. Beyond intensity-based sensing, Light-Emitting Diode (LED) light sources exhibit flickering characteristics with controllable frequencies. Recent studies have begun to exploit the modulation of LED flickering frequencies as an alternative means of inferring users' spatial locations. In this context, LED bulbs with controlled frequency variations can act as virtual landmarks, supporting indoor localization and navigation. Furthermore, prior work has shown that when users capture images of ambient lighting with Complementary Metal-Oxide Semiconductor (CMOS) cameras on mobile devices [ZZ18], the rolling shutter effect, an inherent characteristic of CMOS imaging, appears as striped patterns in the recorded images. This effect provides an additional opportunity for mobility analytics, as it enables the extraction of spatial information from ambient light sources without requiring explicit control over their flickering frequencies. As a result, CMOS-equipped devices can exploit naturally occurring light-based patterns to construct virtual landmarks, thereby supporting more efficient localization and environmental perception in mobility analytics applications.

Future Work on Cross-Domain Sensing Data

In recent years, many mobile devices start to support the measurement of RTT from ambient Wi-Fi APs as an alternative to traditional RSS. Wi-Fi RTT is a Time of Flight (ToF)-based ranging technique, also referred to as FTM, and is standardized in IEEE 802.11mc. In this technique, precisely timed Wi-Fi probe exchanges are performed between users' devices and ambient Wi-Fi APs. Specifically, a Wi-Fi AP first transmits a probe packet to the user device, which then responds by returning the received probe. Since the probe packet records the exact transmission and reception timestamps, the device can estimate its distance to the Wi-Fi AP based on signal propagation speed, thereby providing spatial information for mobility analytics. Compared with Wi-Fi RSS, which is highly susceptible to multipath effects and fading caused by obstacles and reflections, the spatial information provided by RTT is generally more reliable. This is because RTT primarily relies on signal propagation time and is therefore less sensitive to variations in signal strength induced by environmental dynamics. In addition, RTT typically processes only the first arriving signal path, which further helps mitigate distortions introduced by multipath propagation and interference. As a result, while RSS-based localization can be significantly degraded by destructive interference, RTT-based localization remains more robust in complex indoor environments.

However, RSS-based localization also demonstrates clear advantages in many practical scenarios. In particular, RSS-based localization provides higher compatibility and lower deployment cost because it can directly leverage existing Wi-Fi infrastructure. Most Wi-Fi APs already deployed in indoor environments support RSS measurements, allowing localization to be implemented using off-the-shelf devices without additional hardware or infrastructure upgrades. In contrast, RTT-based localization requires IEEE 802.11mc-compliant Wi-Fi APs and user devices, which are not universally available, often necessitating dedicated hardware deployment and increasing system cost.

Recent works [Cho22; SWL+22] provide comparative analyses of RSS and RTT data, demonstrating that the two modalities capture different characteristics of wireless signals and offer complementary information for mobility analytics. As a result, cross-domain utilization of RSS and RTT can improve the reliability of wireless-based mobility analytics. By jointly exploiting the respective advantages of these sensing modalities, a collaborative approach can achieve more accurate, robust, and scalable mobility analytics across diverse application scenarios.

List of Figures

1.1 Mobility analytics based on sensing technology. 2
1.2 Passenger flow estimation based on multiple sensing data. 9
1.3 Presence detection based on multiple sensing data. 10
1.4 Research goals of this dissertation. 12
1.5 Proximity detection based on multi-modal sensing data. 13
1.6 Visual attention detection based on cross-domain sensing data. . 15
1.7 The radio map adaptation in this dissertation refers to 1) generating denser fingerprints for existing APs, and 2) generating radio maps for newly observed APs. 18

3.1 An overview of the proposed system. 28
3.2 Examples of mobility similarity: (a) inter-similarity between two devices' mobility trajectories, and (b) self-similarity along individual mobility trajectories. 31
3.3 An example of inter-similarity computation. 33
3.4 Frequency-domain movement signatures of two devices when they are making movements together. 35
3.5 The user interface of the proposed system. 39
3.6 Two small-scale experiments. 40
3.7 Three mobility scenarios of the two devices. 41
3.8 Detection error rates with different mobility trajectories. 42
3.9 Inter-similarity between d_i and d_j in the library and apartment. 43
3.10 Self-similarity of each device in the library. 45
3.11 Movement similarity between d_i and d_j in the library. 46
3.12 The movement similarities for handshaking and non-handshaking scenarios. 47
3.13 The floor plan of the large-scale environment. 47
3.14 Comprehensive analyses of our similarity metrics vs. co-flow similarity metrics. 48
3.15 Self-similarity of each device in the complex environment. 51
3.16 Inter-similarity between devices in the complex environment. . . 52
3.17 Movement similarity between devices in the complex environment. 54
3.18 Validation with the correlated ground truth. 55
3.19 Accuracy of our system with different time intervals 56
3.20 The impact of different thresholds on system sensitivity, specificity and accuracy. 57
3.21 Analyses of correlations between cyber distances and physical distances. 60
3.22 Time interval I vs. execution time. 62

4.1 System framework. 66
4.2 Creation of visual signatures. 67
4.3 Visual signatures of u_i when u_i views different positions of the monitor. 69
4.4 Illustration of similarity score between visual signatures. 70
4.5 Illustration of attention window. 71
4.6 Detection of visual transitions for user u_i. 72
4.7 Illustration of attention window. 77
4.8 An example of optimal matching. 78
4.9 Prototype of the designed wearable device. 78
4.10 Experimental environments. 79
4.11 Experimental scenarios. 80
4.12 Visual attention similarity between u_i and u_j under different scenarios in the indoor office. 81
4.13 Light distributions on the eyes of users. 82
4.14 System performance in different scenarios. 84
4.15 System performance with different external factors and scenarios. 85
4.16 Large-scale dataset. 88
4.17 Creation of attention traces. 89
4.18 Correlation between visual attention similarity and overlap of attention traces. 90
4.19 Impact of importance level and equality level. 91

5.1 Overview of system framework. 93
5.2 Network architecture. 97
5.3 Creation of fingerprints. 100
5.4 Short Text 101
5.5 Overall performance comparison with SOTA over a year. 105
5.6 Performance decay vs. number of generated APs. 106
5.7 Quality analysis of generated APs. 106
5.8 Performance decay vs. number of generated APs. 108
5.9 Quality analysis of generated APs. 108
5.10 Localization accuracy in different environments. 111
5.11 Advanced study in different environments. 113
5.12 An example of generation with sparse Wi-Fi samples. 114

List of Tables

1.1 Advantages of passive sensing-based mobility analytics. 4

3.1 Performance comparison in two small-scale environments. 46
3.2 Energy and communication costs for different devices. 58

4.1 System performance in different environments. 82
4.2 System performance with various external factors 84
4.3 Performance comparison with other research. 86

5.1 Network architecture of the generator. 103
5.2 Network architecture of the discriminator. 104
5.3 Average localization error (m). 110

List of Algorithms

1 Inter-similarity Quantification $(\widetilde{\mathbb{W}}_i, \widetilde{\mathbb{W}}_j)$. 34

2 Computation of Pairwise Similarity Scores 75

3 Computation of Visual Attention Similarity 76

4 RSS Correlation-based Fingerprinting. 98

Glossary

AI Artificial Intelligence i, 5, 7
AIGC Artificial Intelligence Generated Content 17
AoA Angle of Arrival 4, 24
APIs Application Programming Interfaces 56
APs Access Points 3, 4, 5, 6, 7, 12, 14, 17, 18, 19, 22, 24, 25, 26, 27, 28, 29, 30, 32, 37, 38, 39, 46, 56, 58, 59, 61, 62, 93, 94, 95, 96, 98, 99, 100, 101, 102, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 115, 117, 121, 122, 123, 124

BLE Bluetooth Low Energy 3, 4, 6, 10, 17, 22, 102
BLSTM Binary Long Short-Term Memory 24
BSSID Basic Service Set Identifiers 58

CMOS Complementary Metal-Oxide Semiconductor 121
CNN Convolutional Neural Network 23, 25, 87
CSI Channel state information 4, 6, 25

DCT Discrete Cosine Transform 23
DST Discrete Cosine Transform 23
DTW Dynamic Time Warping 23, 32, 33, 34, 48, 59, 73, 91, 120

FFT Fast Fourier Transform 30, 37
FTM Fine Time Measurement 7, 24, 25, 122

GAN Generative Adversarial Network ii, 17, 18, 19, 25
GPR Gaussian Process Regression ii, 18, 19
GPS Global Positioning System 8, 9, 12, 17

HOOF Histogram of Optical Flow 23

IMU Inertial Measurement Unit i, ii, 3, 4, 6, 8, 9, 12, 14, 17, 25, 54, 55, 117
IoT Internet of Things i, 1, 3, 5, 11

LBP Local Binary Pattern 23
LED Light-Emitting Diode 121
LiDAR Light Detection and Ranging i, 2, 3, 4
LOS Line-of-Sight 8, 12, 17, 21, 29
LSH Locality-Sensitive Hashing 23
LSTM Long Short-Term Memory 24, 25

PCA Principal Component Analysis 23, 24

RADAR Radio Detection And Ranging i, 2, 3, 4, 5

RBT Random Base Transform 23
RFID Radio Frequency Identification 2, 3
RNNs Recurrent Neural Networks 24
RSS Received Signal Strength 3, 4, 5, 6, 7, 10, 14, 17, 18, 19, 27, 28, 29, 30, 32, 37, 46, 50, 58, 59, 61, 62, 93, 94, 95, 96, 98, 99, 100, 102, 106, 107, 109, 110, 114, 115, 121, 122
RTT Round Trip Time 5, 6, 7, 24, 122

SIFT Scale-Invariant Features 23
SOTA State-Of-The-Art 19, 100, 101, 104, 109, 110, 112
SURF Speeded-Up Robust Features 23
SVD Singular Value Decomposition 23

TMK Temporal Match Kernel 23
ToA Time of Arrival 4, 24
ToF Time of Flight 122

VR Virtual Reality 15

Bibliography

[AAC18] A. Achroufene, Y. Amirat, and A. Chibani. "RSS-based indoor localization using belief function theory". In: *IEEE Transactions on Automation Science and Engineering* 16.3 (2018), pp. 1163–1180 (Cited on page 6).

[ABD+23] A. Aubry, P. Babu, A. De Maio, G. Fatima, and N. Sahu. "A robust framework to design optimal sensor locations for TOA or RSS source localization techniques". In: *IEEE Transactions on Signal Processing* 71 (2023), pp. 1293–1306 (Cited on page 3).

[Al-12] M. A. Al-Khedher. "Hybrid GPS-GSM Localization of Automobile Tracking System". In: *Computing Research Repository* abs/1201.2630 (2012) (Cited on page 21).

[AMH18] D. AlShamaa, F. Mourad-Chehade, and P. Honeine. "Mobility-based Tracking Using WiFi RSS in Indoor Wireless Sensor Networks". In: *Int'l Conf. New Technologies, Mobility and Security*. 2018, pp. 1–5 (Cited on page 22).

[ANR74] N. Ahmed, T. Natarajan, and K. R. Rao. "Discrete cosine transform". In: *IEEE Trans. Computers* 100.1 (1974), pp. 90–93 (Cited on page 68).

[AUM+24] T. Ahmad, M. Usman, M. Murtaza, I. B. Benitez, A. Anwar, V. Vassiliou, A. Irshad, X. J. Li, and E. A. Al-Ammar. "A Novel Self-Calibrated UWB Based Indoor Localization Systems for Context-Aware Applications". In: *IEEE Transactions on Consumer Electronics* (2024) (Cited on page 17).

[BCL+16] P. Barsocchi, A. Crivello, D. La Rosa, and F. Palumbo. "A multisource and multivariate dataset for indoor localization methods based on WLAN and geo-magnetic field fingerprinting". In: *IPIN*. 2016, pp. 1–8 (Cited on page 58).

[BTG06] H. Bay, T. Tuytelaars, and L. V. Gool. "Surf: Speeded up robust features". In: *European conference on computer vision*. Springer. 2006, pp. 404–417 (Cited on page 23).

[BXG+20] Y. Bu, L. Xie, Y. Gong, C. Wang, L. Yang, J. Liu, and S. Lu. "RF-dial: Rigid motion tracking and touch gesture detection for interaction via RFID tags". In: *IEEE Transactions on Mobile Computing* 21.3 (2020), pp. 1061–1080 (Cited on page 3).

[BYC+21] L. Bai, Y. Yang, M. Chen, C. Feng, C. Guo, W. Saad, and S. Cui. "Computer vision-based localization with visible light communications". In: *IEEE Transactions on Wireless Communications* 21.3 (2021), pp. 2051–2065 (Cited on page 1).

[CC20] K. M. Chen and R. Y. Chang. "Semi-supervised learning with GANs for device-free fingerprinting indoor localization". In: *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE. 2020, pp. 1–6 (Cited on page 25).

[CCL15] C.-L. Chou, H.-T. Chen, and S.-Y. Lee. "Pattern-based near-duplicate video retrieval and localization on web-scale videos". In: *IEEE Transactions on Multimedia* 17.3 (2015), pp. 382–395 (Cited on page 23).

[ÇDG+18] B. S. Çiftler, S. Dikmese, Í. Güvenç, K. Akkaya, and A. Kadri. "Occupancy counting with burst and intermittent signals in smart buildings". In: *IEEE Internet of Things Journal* 5.2 (2018), pp. 724–735 (Cited on page 22).

[CFL+22] G. Cerro, L. Ferrigno, M. Laracca, G. Miele, F. Milano, and V. Pingerna. "Uwb-based indoor localization: How to optimally design the operating setup?" In: *IEEE Transactions on Instrumentation and Measurement* 71 (2022), pp. 1–12 (Cited on page 17).

[Cho22] J. Choi. "Enhanced Wi-Fi RTT ranging: A sensor-aided learning approach". In: *IEEE Transactions on Vehicular Technology* 71.4 (2022), pp. 4428–4437 (Cited on pages 24 sq., 122).

[CKK21] J.-H. Choi, J.-E. Kim, and K.-T. Kim. "Deep learning approach for radar-based people counting". In: *IEEE Internet of Things Journal* 9.10 (2021), pp. 7715–7730 (Cited on page 3).

[CLC+20] J. Choi, G. Lee, S. Choi, and S. Bahk. "Smartphone based indoor path estimation and localization without human intervention". In: *IEEE Transactions on Mobile Computing* 21.2 (2020), pp. 681–695 (Cited on page 17).

[CLZ+22] Y. Cheng, C. Li, Y. Zhang, S. He, and J. Chen. "Spatial–temporal urban mobility pattern analysis during covid-19 pandemic". In: *IEEE Transactions on Computational Social Systems* 11.1 (2022), pp. 38–50 (Cited on page 1).

[CSM06] B. Coskun, B. Sankur, and N. Memon. "Spatio–temporal transform based video hashing". In: *ieee Transactions on Multimedia* 8.6 (2006), pp. 1190–1208 (Cited on page 23).

[CWB+16] E. Cheung, T. K. Wong, A. Bera, X. G. Wang, and D. Manocha. "LCrowdV: Generating Labeled Videos for Simulation-Based Crowd Behavior Learning". In: *European Conference on Computer Vision*. 2016, pp. 709–727 (Cited on page 21).

[CXW+19] Y. Cui, H. Xu, J. Wu, Y. Sun, and J. Zhao. "Automatic vehicle tracking with roadside LiDAR data for the connected-vehicles system". In: *IEEE Intelligent Systems* 34.3 (2019), pp. 44–51 (Cited on page 3).

[DL18] F. Dechterenko and J. Lukavsky. "Robustness of metrics used for scanpath comparison". In: *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*. 2018, pp. 1–5 (Cited on page 33).

[DNX+18] J. Dong, M. Noreikis, Y. Xiao, and A. Ylä-Jääski. "ViNav: A vision-based indoor navigation system for smartphones". In: *IEEE Transactions on Mobile Computing* 18.6 (2018), pp. 1461–1475 (Cited on page 17).

[DTC+21] F. Demrozi, C. Turetta, F. Chiarani, P. H. Kindt, and G. Pravadelli. "Estimating indoor occupancy through low-cost BLE devices". In: *IEEE Sensors Journal* 21.15 (2021), pp. 17053–17063 (Cited on page 4).

[DWB19] J. Dalins, C. Wilson, and D. Boudry. "PDQ & TMK+ PDQF–A Test Drive of Facebook's Perceptual Hashing Algorithms". In: *arXiv preprint arXiv:1912.07745* (2019) (Cited on pages 23, 86).

[DYY+17] H. Du, Z. Yu, F. Yi, Z. Wang, Q. Han, and B. Guo. "Recognition of group mobility level and group structure with mobile devices". In: *IEEE Trans. Mobile Computing* 17.4 (2017), pp. 884–897 (Cited on page 22).

[EFW10] M. M. Esmaeili, M. Fatourechi, and R. K. Ward. "A robust and fast video copy detection system using content-based fingerprinting". In: *IEEE Transactions on information forensics and security* 6.1 (2010), pp. 213–226 (Cited on page 23).

[Fan86] B. T. Fang. "Trilateration and extension to global positioning system navigation". In: *Journal of Guidance, Control, and Dynamics* 9.6 (1986), pp. 715–717 (Cited on page 21).

[GA19] S. Z. Gurbuz and M. G. Amin. "Radar-based human-motion recognition with deep learning: Promising applications for indoor monitoring". In: *IEEE Signal Processing Magazine* 36.4 (2019), pp. 16–28 (Cited on page 3).

[GAA+17] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. "Improved training of wasserstein gans". In: *Advances in neural information processing systems* 30 (2017) (Cited on page 96).

[GAK+20] I. Grishchenko, A. Ablavatski, Y. Kartynnik, K. Raveendran, and M. Grundmann. "Attention mesh: High-fidelity face mesh prediction in real-time". In: *arXiv preprint arXiv:2006.10962* (2020) (Cited on page 67).

[GCR12] W. Ge, R. T. Collins, and R. B. Ruback. "Vision-Based Analysis of Small Groups in Pedestrian Crowds". In: *IEEE Trans. Pattern Analysis and Machine Intelligence* 34.5 (2012), pp. 1003–1016 (Cited on page 21).

[GIM+99] A. Gionis, P. Indyk, R. Motwani, et al. "Similarity search in high dimensions via hashing". In: *Vldb*. Vol. 99. 6. 1999, pp. 518–529 (Cited on page 23).

[GJ23] W. Guo and L. Jing. "Toward low-cost passive motion tracking with one pair of commodity Wi-Fi devices". In: *IEEE Journal of Indoor and Seamless Positioning and Navigation* 1 (2023), pp. 39–52 (Cited on page 4).

[GLL+23] R. Gao, W. Li, J. Liu, S. Dai, M. Zhang, L. Wang, and D. Zhang. "Wicgesture: Meta-motion based continuous gesture recognition with wi-fi". In: *IEEE Internet of Things Journal* (2023) (Cited on page 4).

[Gov20] S. Government Technology Agency. *TraceTogether*. https://www.tracetogether.gov.sg/. 2020 (Cited on page 12).

[GRN+24] S. García-de-Villa, L. R. Ruiz, G. G.-V. Neira, M. N. Álvarez, E. Huertas-Hoyas, A. J. Del-Ama, M. C. Rodriguez-Sanchez, F. Seco, and A. R. Jiménez. "Validation of an IMU-based gait analysis method for assessment of fall risk against traditional methods". In: *IEEE journal of biomedical and health informatics* (2024) (Cited on page 4).

[HLC+05] S. Hong, M. H. Lee, H.-H. Chun, S.-H. Kwon, and J. L. Speyer. "Observability of error States in GPS/INS integration". In: *IEEE Trans. Vehicular Technology* 54.2 (2005), pp. 731–743 (Cited on page 21).

[HS+88] C. Harris, M. Stephens, et al. "A combined corner and edge detector". In: *Alvey vision conference*. Vol. 15. 50. Citeseer. 1988, pp. 10–5244 (Cited on page 23).

[HS19] S. He and K. G. Shin. "Crowd-Flow Graph Construction and Identification with Spatio-Temporal Signal Feature Fusion". In: *IEEE INFOCOM*. 2019, pp. 757–765 (Cited on pages 22, 39, 48).

[HW21] Y. Huang and F.-J. Wu. "CRISIS: Cyber-physical social distancing based on multi-modal data from mobile devices". In: *IEEE Transactions on Mobile Computing* 22.5 (2021), pp. 2551–2568 (Cited on page 20).

[HW25a] Y. Huang and F.-J. Wu. "Spatial Reconstruction and Localization based on Radio Map Adaption to Time-varying Environments". In: *(Preparing for submission)* (2025) (Cited on page 20).

[HW25b] Y. Huang and F.-J. Wu. "V-Groups: Matching Light Traces on Human Eyes for Detecting Visual Attention Groups". In: *(Preparing for submission)* (2025) (Cited on page 20).

[HWH+20] Y. Huang, F.-J. Wu, C. Hakert, G. von der Brüggen, K.-H. Chen, J.-J. Chen, P. Böcker, P. Chernikov, L. Cruz, Z. Duan, et al. "Demo Abstract: Perception vs. Reality-Never Believe in What You See". In: *2020 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE. 2020, pp. 363–364 (Cited on pages 10, 20).

[HXH+18] Y. Hu, Y. Xiong, W. Huang, X.-Y. Li, P. Yang, Y. Zhang, and X. Mao. "Lightitude: Indoor positioning using uneven light intensity distribution". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2.2 (2018), pp. 1–25 (Cited on page 17).

[HYD+19] M. T. Hoang, B. Yuen, X. Dong, T. Lu, R. Westendorp, and K. Reddy. "Recurrent neural networks for accurate RSSI indoor localization". In: *IEEE Internet of Things Journal* 6.6 (2019), pp. 10639–10651 (Cited on pages 17, 101).

[HYH20] O. Hashem, M. Youssef, and K. A. Harras. "WiNar: RTT-based sub-meter indoor localization using commercial devices". In: *2020 IEEE international conference on pervasive computing and communications (PerCom)*. IEEE. 2020, pp. 1–10 (Cited on page 6).

[HZR+16] K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778 (Cited on page 97).

[JHQ+16] Z. Jiang, J. Han, C. Qian, W. Xi, K. Zhao, H. Ding, S. Tang, J. Zhao, and P. Yang. "VADS: Visual attention detection with a smartphone". In: *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE. 2016, pp. 1–9 (Cited on page 15).

[JKL18] Y.-H. Jin, K.-W. Ko, and W.-H. Lee. "An Indoor Location-Based Positioning System Using Stereo Vision with the Drone Camera". In: *Mobile Information Systems* 2018.1 (2018), p. 5160543 (Cited on page 6).

[JLT+18] J. Jiao, F. Li, W. Tang, Z. Deng, and J. Cao. "A hybrid fusion of wireless signals and RGB image for indoor positioning". In: *International Journal of Distributed Sensor Networks* 14.2 (2018), p. 1550147718757664 (Cited on page 6).

[JP24] S. A. Junoh and J.-Y. Pyun. "Augmentation of Fingerprints for Indoor BLE Localization Using Conditional GANs". In: *IEEE Access* (2024) (Cited on pages 18, 25).

[JSC20] F. Jin, A. Sengupta, and S. Cao. "mmfall: Fall detection using 4-d mmwave radar and a hybrid variational rnn autoencoder". In: *IEEE Transactions on Automation Science and Engineering* 19.2 (2020), pp. 1245–1257 (Cited on page 1).

[KB17] F. Khelifi and A. Bouridane. "Perceptual video hashing for content identification and authentication". In: *IEEE Transactions on Circuits and Systems for Video Technology* 29.1 (2017), pp. 50–67 (Cited on page 23).

[KPP+17] G. Kordopatis-Zilos, S. Papadopoulos, I. Patras, and Y. Kompatsiaris. "Near-duplicate video retrieval by aggregating intermediate cnn layers". In: *International conference on multimedia modeling*. Springer. 2017, pp. 251–263 (Cited on page 24).

[KPP+19] G. Kordopatis-Zilos, S. Papadopoulos, I. Patras, and I. Kompatsiaris. "Visil: Fine-grained spatio-temporal video similarity learning". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 6351–6360 (Cited on pages 23, 86).

[KRA+20] T. Kalsoom, N. Ramzan, S. Ahmed, and M. Ur-Rehman. "Advances in sensor technologies in the era of smart factory and industry 4.0". In: *Sensors* 20.23 (2020), p. 6783 (Cited on page 1).

[KSH+04] Y. Ke, R. Sukthankar, L. Huston, Y. Ke, and R. Sukthankar. "Efficient near-duplicate detection and sub-image retrieval". In: *ACM multimedia*. Vol. 4. 1. Citeseer. 2004, p. 5 (Cited on page 23).

[LBG+06] J. Law-To, O. Buisson, V. Gouet-Brunet, and N. Boujemaa. "Robust voting algorithm based on labels of behavior for video copy detection". In: *Proceedings of the 14th ACM international conference on Multimedia*. 2006, pp. 835–844 (Cited on page 23).

[LCL+19] S. Li, Z. Chen, X. Li, J. Lu, and J. Zhou. "Unsupervised variational video hashing with 1D-CNN-LSTM networks". In: *IEEE Transactions on Multimedia* 22.6 (2019), pp. 1542–1554 (Cited on page 24).

[LCW+23] W. Li, R. Chen, Y. Wu, and H. Zhou. "Indoor positioning system using a single-chip millimeter wave radar". In: *IEEE Sensors Journal* 23.5 (2023), pp. 5232–5242 (Cited on page 3).

[LFZ+16] P. Luo, Y. Fei, L. Zhang, and A. A. Ding. "Differential fault analysis of SHA3-224 and SHA3-256". In: *2016 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*. IEEE. 2016, pp. 4–15 (Cited on page 28).

[LGY+21] X. Liu, L. Guo, H. Yang, and X. Wei. "Visible Light Positioning Based on Collaborative LEDs and Edge Computing". In: *IEEE Transactions on Computational Social Systems* 9.1 (2021), pp. 324–335 (Cited on page 17).

[LH12] M. Lee and D. Han. "Voronoi tessellation based interpolation method for Wi-Fi radio map construction". In: *IEEE Communications Letters* 16.3 (2012), pp. 404–407 (Cited on page 17).

[LHY+18] X. Liang, Z. Huang, S. Yang, and L. Qiu. "Device-free motion & trajectory detection via RFID". In: *ACM Transactions on Embedded Computing Systems (TECS)* 17.4 (2018), pp. 1–27 (Cited on page 3).

[LJW+21] C. H. Lam, K. E. Jeon, S. Wong, and J. She. "Distance estimation using BLE beacon on stationary and mobile objects". In: *IEEE Internet of Things Journal* 9.7 (2021), pp. 4928–4939 (Cited on page 4).

[LLT+16] V. E. Liong, J. Lu, Y.-P. Tan, and J. Zhou. "Deep video hashing". In: *IEEE Transactions on Multimedia* 19.6 (2016), pp. 1209–1219 (Cited on page 24).

[LLX12] H. Liu, H. Lu, and X. Xue. "A segmentation and graph-based video sequence matching method for video copy detection". In: *IEEE transactions on knowledge and data engineering* 25.8 (2012), pp. 1706–1718 (Cited on page 23).

[LM12] M. Li and V. Monga. "Robust video hashing via multilinear subspace projections". In: *IEEE transactions on image processing* 21.10 (2012), pp. 4397–4409 (Cited on page 23).

[LMS+18] Y. Lu, A. Misra, W. Sun, and H. Wu. "Smartphone Sensing Meets Transport Data: A Collaborative Framework for Transportation Service Analytics". In: *IEEE Trans. Mobile Computing* 17.4 (2018), pp. 945–960 (Cited on page 11).

[Low99] D. G. Lowe. "Object recognition from local scale-invariant features". In: *Proceedings of the seventh IEEE international conference on computer vision*. Vol. 2. Ieee. 1999, pp. 1150–1157 (Cited on page 23).

[LP17] D. A. Levin and Y. Peres. *Markov chains and mixing times*. Vol. 107. American Mathematical Soc., 2017 (Cited on page 23).

[LQL+19] Q. Li, H. Qu, Z. Liu, N. Zhou, W. Sun, S. Sigg, and J. Li. "AF-DCGAN: Amplitude feature deep convolutional GAN for fingerprint construction in indoor localization systems". In: *IEEE Transactions on Emerging Topics in Computational Intelligence* 5.3 (2019), pp. 468–480 (Cited on page 25).

[LRC19] E. Longo, A. Redondi, and M. Cesana. "Accurate occupancy estimation with WiFi and bluetooth/BLE packet capture". In: *Computer Networks* 163.9 (2019) (Cited on page 22).

[LXY+24] D. Li, J. Xu, Z. Yang, and C. Tang. "Train Once, Locate Anytime for Anyone: Adversarial Learning-based Wireless Localization". In: *ACM Transactions on Sensor Networks* 20.2 (2024), pp. 1–21 (Cited on pages 17, 25, 104).

[LYH+22] Y. Lin, K. Yu, L. Hao, J. Wang, and J. Bu. "An indoor Wi-Fi localization algorithm using ranging model constructed with transformed RSSI and BP neural network". In: *IEEE Transactions on Communications* 70.3 (2022), pp. 2163–2177 (Cited on page 25).

[LYK+21] R. Levie, Ç. Yapar, G. Kutyniok, and G. Caire. "RadioUNet: Fast radio map estimation with convolutional neural networks". In: *IEEE Transactions on Wireless Communications* 20.6 (2021), pp. 4001–4015 (Cited on pages 17, 25).

[MA15] P. Malaji and S. Ali. "Analysis of energy harvesting from multiple pendulums with and without mechanical coupling". In: *The European Physical Journal Special Topics* 224.14-15 (2015), pp. 2823–2838 (Cited on page 30).

[MC21] K. Min and J. J. Corso. "Integrating human gaze into attention for egocentric activity recognition". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2021, pp. 1069–1078 (Cited on page 15).

[MMS+17] S. Memon, M. M. Memon, F. K. Shaikh, and S. Laghari. "Smart indoor positioning using BLE technology". In: *IEEE Int'l Conf. Engineering Technologies and Applied Sciences*. 2017, pp. 1–5 (Cited on page 22).

[MPG+10] M. Moussaïd, N. Perozo, S. Garnier, D. Helbing, and G. Theraulaz. "The walking behaviour of pedestrian social groups and its impact on crowd dynamics". In: *PloS one* 5.4 (2010), e10047 (Cited on page 21).

[MRT+18] G. M. Mendoza-Silva, P. Richter, J. Torres-Sospedra, E. S. Lohan, and J. Huerta. "Long-term WiFi fingerprinting dataset for research on robust indoor positioning". In: *Data* 3.1 (2018), p. 3 (Cited on page 101).

[MRY+24] M. Mohsen, H. Rizk, H. Yamaguchi, and M. Youssef. "TimeSense: Multi-Person Device-free Indoor Localization via RTT". In: *IEEE Internet of Things Journal* (2024) (Cited on page 6).

[MS23] S. A. Mousavi and R. Selmic. "Wearable smart rings for multi-finger gesture recognition using supervised learning". In: *IEEE Transactions on Instrumentation and Measurement* (2023) (Cited on page 4).

[MS24] N. C. Matson and K. Sundaresan. "Online Radio Environment Map Creation via UAV Vision for Aerial Networks". In: *IEEE INFOCOM 2024-IEEE Conference on Computer Communications*. IEEE. 2024, pp. 81–90 (Cited on pages 17, 25).

[MSB+21] Z. Mahrez, E. Sabir, E. Badidi, W. Saad, and M. Sadik. "Smart urban mobility: When mobility systems meet smart data". In: *IEEE Transactions on Intelligent Transportation Systems* 23.7 (2021), pp. 6222–6239 (Cited on page 1).

[MWP+20] C. Ma, B. Wu, S. Poslad, and D. R. Selviah. "Wi-Fi RTT ranging performance characterization and positioning system design". In: *IEEE Transactions on Mobile Computing* 21.2 (2020), pp. 740–756 (Cited on page 24).

[NCC+21] W. Njima, M. Chafii, A. Chorti, R. M. Shubair, and H. V. Poor. "Indoor localization using data augmentation via selective generative adversarial networks". In: *IEEE access* 9 (2021), pp. 98337–98347 (Cited on pages 18, 25).

[PCM+22] S. Paiva, V. Corcoba, F. Mourão, X. G. Pañeda, D. Melendi, and R. García. "Analysis of mobility changes caused by COVID-19 in a context of moderate restrictions using data collected by mobile devices". In: *IEEE access* 10 (2022), pp. 8906–8915 (Cited on page 1).

[QTN+22] D. Quezada-Gaibor, J. Torres-Sospedra, J. Nurmi, Y. Koucheryavy, and J. Huerta. "SURIMI: Supervised radio map augmentation with deep learning and a generative adversarial network for fingerprint-based indoor positioning". In: *2022 IEEE 12th International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE. 2022, pp. 1–8 (Cited on pages 18, 25, 104).

[RCW+12] J. Ren, F. Chang, T. Wood, and J. R. Zhang. "Efficient video copy detection via aligning video signature time series". In: *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval*. 2012, pp. 1–8 (Cited on pages 23, 86).

[RFB15] O. Ronneberger, P. Fischer, and T. Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer. 2015, pp. 234–241 (Cited on page 96).

[RHZ+21] H. Rong, S. Huo, Q. Zhang, H. Zheng, and C. Yang. "GREEN: A global optimization scheme for transportation efficiency by mining taxi mobility". In: *IEEE Transactions on Intelligent Transportation Systems* 23.2 (2021), pp. 1596–1606 (Cited on page 1).

[RR13] R. Roopalakshmi and G. R. M. Reddy. "A novel spatio-temporal registration framework for video copy localization based on multimodal features". In: *Signal processing* 93.8 (2013), pp. 2339–2351 (Cited on page 23).

[SBB+15] C. Sarraute, J. Brea, J. Burroni, and P. Blanc. "Inference of demographic attributes based on mobile phone usage patterns and social network topology". In: *Social Network Analysis and Mining* 5.1 (2015), pp. 39–39 (Cited on page 12).

[SCC16] F. Solera, S. Calderara, and R. Cucchiara. "Socially Constrained Structural Learning for Groups Detection in Crowd". In: *IEEE Trans. Pattern Analysis and Machine Intelligence* 38.5 (2016), pp. 995–1008 (Cited on page 21).

[SCL+18] J. Shen, J. Cao, X. Liu, and S. Tang. "SNOW: Detecting Shopping Groups Using WiFi". In: *IEEE Internet of Things Journal* 5.5 (2018), pp. 3908–3917 (Cited on page 22).

[SHZ+15] Y. Shu, Y. Huang, J. Zhang, P. Coué, P. Cheng, J. Chen, and K. G. Shin. "Gradient-based fingerprinting for indoor localization and tracking". In: *IEEE Transactions on Industrial Electronics* 63.4 (2015), pp. 2424–2433 (Cited on page 17).

[SLJ+14] R. Sen, Y. Lee, K. Jayarajah, A. Misra, and R. K. Balan. "Grumon: Fast and accurate group monitoring for heterogeneous urban spaces". In: *Proceedings of the 12th ACM conference on embedded network sensor systems*. 2014, pp. 46–60 (Cited on page 17).

[SMB+17] H. Senaratne, M. Mueller, M. Behrisch, F. Lalanne, J. Bustos-Jiménez, J. Schneidewind, D. Keim, and T. Schreck. "Urban mobility analysis with mobile network data: A visual analytics approach". In: *IEEE Transactions on Intelligent Transportation Systems* 19.5 (2017), pp. 1537–1546 (Cited on page 1).

[SMS+18] J. Steil, P. Müller, Y. Sugano, and A. Bulling. "Forecasting user attention during everyday mobile interactions using device-integrated and wearable sensors". In: *Proceedings of the 20th international conference on human-computer interaction with mobile devices and services*. 2018, pp. 1–13 (Cited on page 15).

[STJ+17] L. E. Simona, Torres-Sospedra, Joaquín, L. Helena, R. Philipp, P. Zhe, and H. Joaquín. "Wi-Fi crowdsourced fingerprinting dataset for indoor positioning". In: *Data* 2.4 (2017), p. 32 (Cited on page 58).

[SW17] G. Solmaz and F.-J. Wu. "Together or alone: Detecting group mobility with wireless fingerprints". In: *IEEE Int'l Conf. Comm.* 2017, pp. 1–7 (Cited on page 22).

[SWL+22] M. Sun, Y. Wang, K. Liu, C. De Cock, W. Joseph, and D. Plets. "Smartphone-based WiFi FTM Fingerprinting Approach with Map-aided Particle Filter". In: *2022 IEEE 12th International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE. 2022, pp. 1–8 (Cited on pages 17, 25, 122).

[SWX+20] S. Shi, L. Wang, S. Xu, and X. Wang. "Prediction of intra-urban human mobility by integrating regional functions and trip intentions". In: *IEEE Transactions on Knowledge and Data Engineering* 34.10 (2020), pp. 4972–4981 (Cited on page 1).

[SYW+10] L. Shang, L. Yang, F. Wang, K.-P. Chan, and X.-S. Hua. "Real-time large scale near-duplicate web video retrieval". In: *Proceedings of the 18th ACM international conference on Multimedia*. 2010, pp. 531–540 (Cited on page 23).

[SZL+18] J. Song, H. Zhang, X. Li, L. Gao, M. Wang, and R. Hong. "Self-supervised video hashing with hierarchical binary auto-encoder". In: *IEEE Transactions on Image Processing* 27.7 (2018), pp. 3210–3221 (Cited on page 24).

[TBB+23] B. C. Tedeschini, M. Brambilla, L. Barbieri, G. Balducci, and M. Nicoli. "Cooperative lidar sensing for pedestrian detection: Data association based on message passing neural networks". In: *IEEE Transactions on Signal Processing* (2023) (Cited on page 3).

[TLT+21] X. Tong, H. Li, X. Tian, and X. Wang. "Wi-Fi localization enabling self-calibration". In: *IEEE/ACM Transactions on Networking* 29.2 (2021), pp. 904–917 (Cited on page 24).

[TR21] Y. Teganya and D. Romero. "Deep completion autoencoders for radio map estimation". In: *IEEE Transactions on Wireless Communications* 21.3 (2021), pp. 1710–1724 (Cited on page 17).

[TWL+20] X. Tong, Y. Wan, Q. Li, X. Tian, and X. Wang. "CSI fingerprinting localization with low human efforts". In: *IEEE/ACM Transactions on Networking* 29.1 (2020), pp. 372–385 (Cited on pages 17, 25).

[TWL+21] X. Tong, H. Wang, X. Liu, and W. Qu. "MapFi: Autonomous mapping of Wi-Fi infrastructure for indoor localization". In: *IEEE transactions on mobile computing* 22.3 (2021), pp. 1566–1580 (Cited on page 24).

[TZ21] Y. Tao and L. Zhao. "AIPS: An accurate indoor positioning system with fingerprint map adaptation". In: *IEEE Internet of Things Journal* 9.4 (2021), pp. 3062–3073 (Cited on page 17).

[Una20] Unacast. *Social Distancing Scoreboard*. https://www.unacast.com/covid19/social-distancing-scoreboard. 2020 (Cited on page 12).

[VGQ16] V. Varshney, R. K. Goel, and M. A. Qadeer. "Indoor positioning system using Wi-Fi & bluetooth low energy technology". In: *Int'l Conf. Wireless and Optical Comm. Networks*. 2016, pp. 1–6 (Cited on page 22).

[Wel91] E. Welzl. "Smallest enclosing disks (balls and ellipsoids)". In: *New results and new trends in computer science*. Springer, 1991, pp. 359–370 (Cited on page 67).

[WHD+20] F.-J. Wu, Y. Huang, L. Döring, S. Althoff, K. Bitterschulte, K. Y. Chai, L. Mao, D. Grabarczyk, and E. Kovacs. "PassengerFlows: A correlation-based passenger estimator in automated public transport". In: *IEEE Transactions on Network Science and Engineering* 7.4 (2020), pp. 2167–2181 (Cited on pages 8, 20).

[WHN07] X. Wu, A. G. Hauptmann, and C.-W. Ngo. "Practical elimination of near-duplicates from web video search". In: *Proceedings of the 15th ACM international conference on Multimedia*. 2007, pp. 218–227 (Cited on page 23).

[WKT11] F.-J. Wu, Y.-F. Kao, and Y.-C. Tseng. "From wireless sensor networks towards cyber physical systems". In: *Pervasive and Mobile Computing* 7.4 (2011), pp. 397–413 (Cited on page 12).

[WL20] F.-J. Wu and T. Luo. "CrowdPrivacy: Publish More Useful Data with Less Privacy Exposure in Crowdsourced Location-Based Services". In: *ACM Trans. Privacy and Security* 23.1 (2020), pp. 1–25 (Cited on page 12).

[WLC+18] C. Wang, J. Liu, Y. Chen, H. Liu, L. Xie, W. Wang, B. He, and S. Lu. "Multi-touch in the air: Device-free finger tracking and gesture recognition via COTS RFID". In: *IEEE INFOCOM 2018-IEEE conference on computer communications*. IEEE. 2018, pp. 1691–1699 (Cited on page 3).

[WLQ+20] X. Wang, J. Liu, T. Qiu, C. Mu, C. Chen, and P. Zhou. "A real-time collision prediction mechanism with deep learning for intelligent transportation system". In: *IEEE transactions on vehicular technology* 69.9 (2020), pp. 9497–9508 (Cited on page 1).

[WNS+19] Y. Wang, X. Nie, Y. Shi, X. Zhou, and Y. Yin. "Attention-based video hashing for large-scale video retrieval". In: *IEEE Transactions on Cognitive and Developmental Systems* 13.3 (2019), pp. 491–502 (Cited on page 24).

[WS17] W. Wang and J. Shen. "Deep visual attention prediction". In: *IEEE Transactions on Image Processing* 27.5 (2017), pp. 2368–2378 (Cited on page 15).

[WS18] F.-J. Wu and G. Solmaz. "CrowdEstimator: Approximating Crowd Sizes with Multi-modal Data for Internet-of-Things Services". In: *ACM Int'l Conf. on Mobile Systems, Applications, and Services*. 2018, pp. 337–349 (Cited on page 22).

[Wu18] F.-J. Wu. "A Sensor-assisted Emergency Guiding System: Sensor-centric or User-centric?" In: *IEEE Trans. Vehicular Technology* 67.2 (2018), pp. 1598–1611 (Cited on page 12).

[WWM18] X. Wang, X. Wang, and S. Mao. "Deep convolutional neural networks for indoor localization with CSI images". In: *IEEE Transactions on Network Science and Engineering* 7.1 (2018), pp. 316–327 (Cited on page 6).

[WYL+12] C. Wu, Z. Yang, Y. Liu, and W. Xi. "WILL: Wireless indoor localization without site survey". In: *IEEE Trans. Parallel and Distributed Systems* 24.4 (2012), pp. 839–848 (Cited on page 22).

[WYW+21] W. Wei, J. Yan, L. Wan, C. Wang, G. Zhang, and X. Wu. "Enriching indoor localization fingerprint using a single AC-GAN". In: *2021 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE. 2021, pp. 1–6 (Cited on page 25).

[XCZ08] B. Xiao, H. Chen, and S. Zhou. "Distributed Localization Using a Moving Beacon in Wireless Sensor Networks". In: *IEEE Trans. Parallel and Distributed Systems* 19.5 (2008), pp. 587–600 (Cited on page 21).

[XYL+13] L. Xiao, Q. Yan, W. Lou, G. Chen, and Y. T. Hou. "Proximity-based security techniques for mobile users in wireless networks". In: *IEEE Trans. Information Forensics and security* 8.12 (2013), pp. 2089–2100 (Cited on page 12).

[YC09] M.-C. Yeh and K.-T. Cheng. "Video copy detection by fast sequence matching". In: *Proceedings of the ACM International Conference on Image and Video Retrieval*. 2009, pp. 1–7 (Cited on page 23).

[YCC+21] Y. Yu, R. Chen, L. Chen, X. Zheng, D. Wu, W. Li, and Y. Wu. "A novel 3-D indoor localization algorithm based on BLE and multiple sensors". In: *IEEE Internet of Things Journal* 8.11 (2021), pp. 9359–9372 (Cited on page 17).

[YCJ12] G. Yang, N. Chen, and Q. Jiang. "A robust hashing algorithm based on SURF for video copy detection". In: *Computers & Security* 31.1 (2012), pp. 33–39 (Cited on page 23).

[YCS+22] Y. Yu, R. Chen, W. Shi, and L. Chen. "Precise 3D indoor localization and trajectory optimization based on sparse Wi-Fi FTM anchors and built-in sensors". In: *IEEE Transactions on Vehicular Technology* 71.4 (2022), pp. 4042–4056 (Cited on pages 24 sq.).

[YDV+13] S. Yang, P. Dessai, M. Verma, and M. Gerla. "FreeLoc: Calibration-free crowdsourced indoor localization". In: *2013 Proceedings IEEE INFOCOM*. IEEE. 2013, pp. 2481–2489 (Cited on pages 17, 22).

[YHS+09] S.-C. Yeh, W.-H. Hsu, M.-Y. Su, C.-H. Chen, and K.-H. Liu. "A study on outdoor positioning technology using GPS and WiFi networks". In: *IEEE Int'l Conf. Networking, Sensing and Control*. 2009, pp. 597–601 (Cited on page 21).

[YLL09] Z. Yang, Y. Liu, and X.-Y. Li. "Beyond trilateration: On the localizability of wireless ad-hoc networks". In: *IEEE INFOCOM*. 2009, pp. 2392–2400 (Cited on page 21).

[YMC+23] N. Yu, X. Ma, X. Chen, R. Feng, and Y. Wu. "High-Precision Indoor Positioning Method Based on Multi-Feature Fusion of Inertial Sensor Network". In: *IEEE Transactions on Instrumentation and Measurement* (2023) (Cited on page 17).

[You01] S. S. Young. *Computerized data acquisition and analysis for the life sciences: a hands-on guide*. Cambridge University Press, 2001 (Cited on page 38).

[YSR22] D.-h. Yoo, G. Shan, and B.-h. Roh. "A vision-based indoor positioning systems utilizing computer aided design drawing". In: *Proceedings of the 28th Annual International Conference on Mobile Computing and Networking*. 2022, pp. 880–882 (Cited on page 17).

[YWW+21] Z. Yang, R. Wang, D. Wu, H. Wang, H. Song, and X. Ma. "Local trajectory privacy protection in 5G enabled industrial intelligent logistics". In: *IEEE Transactions on Industrial Informatics* 18.4 (2021), pp. 2868–2876 (Cited on page 1).

[YYT22] H.-C. Yen, L.-Y. O. Yang, and Z.-M. Tsai. "3-D indoor localization and identification through RSSI-based angle of arrival estimation with real Wi-Fi signals". In: *IEEE Transactions on Microwave Theory and Techniques* 70.10 (2022), pp. 4511–4527 (Cited on page 3).

[YZX21] X. Yi, Y. Zhou, and F. Xu. "Transpose: Real-time 3d human translation and pose estimation with six inertial sensors". In: *ACM Transactions On Graphics (TOG)* 40.4 (2021), pp. 1–13 (Cited on page 4).

[YZZ22] Z. Yang, Y. Zhang, and Q. Zhang. "Rethinking fall detection with Wi-Fi". In: *IEEE Transactions on Mobile Computing* 22.10 (2022), pp. 6126–6143 (Cited on page 4).

[ZAP+23] X. Zhao, Z. An, Q. Pan, and L. Yang. "Nerf2: Neural radio-frequency radiance fields". In: *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*. 2023, pp. 1–15 (Cited on pages 18, 25).

[ZCL+20] H. Zou, C.-L. Chen, M. Li, J. Yang, Y. Zhou, L. Xie, and C. J. Spanos. "Adversarial learning-enabled automatic WiFi indoor radio map construction and adaptation with mobile robot". In: *IEEE Internet of Things Journal* 7.8 (2020), pp. 6946–6954 (Cited on page 25).

[ZLL+20] X. Zhu, Y. Luo, A. Liu, W. Tang, and M. Z. A. Bhuiyan. "A deep learning-based mobile crowdsensing scheme by predicting vehicle mobility". In: *IEEE Transactions on Intelligent Transportation Systems* 22.7 (2020), pp. 4648–4659 (Cited on page 1).

[ZLL+24] C. Zhu, L. Luo, R. Li, J. Guo, and Q. Wang. "Wearable Motion Analysis System for Thoracic Spine Mobility with Inertial Sensors". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* (2024) (Cited on page 4).

[ZLT+21] M. Zhou, Y. Li, M. J. Tahir, X. Geng, Y. Wang, and W. He. "Integrated statistical test of signal distributions and access point contributions for Wi-Fi indoor localization". In: *IEEE Transactions on Vehicular Technology* 70.5 (2021), pp. 5057–5070 (Cited on page 25).

[ZLX+19] J. Zhao, Y. Li, H. Xu, and H. Liu. "Probabilistic prediction of pedestrian crossing intention using roadside LiDAR data". In: *IEEE Access* 7 (2019), pp. 93781–93790 (Cited on page 3).

[ZQZ+22] X. Zhu, W. Qu, X. Zhou, L. Zhao, Z. Ning, and T. Qiu. "Intelligent Fingerprint-Based Localization Scheme Using CSI Images for Internet of Things". In: *IEEE Transactions on Network Science and Engineering* 9.4 (2022), pp. 2378–2391 (Cited on pages 17, 25).

[ZSZ+24] X. Zhang, W. Sun, J. Zheng, A. Lin, J. Liu, and S. S. Ge. "Wi-Fi-Based Indoor Localization with Interval Random Analysis and Improved Particle Swarm Optimization". In: *IEEE Transactions on Mobile Computing* (2024) (Cited on page 25).

[ZTC+22] C. Zakaria, A. Trivedi, E. Cecchet, M. Chee, P. Shenoy, and R. Balan. "Analyzing the impact of Covid-19 control policies on campus occupancy and mobility via wifi sensing". In: *ACM Transactions on Spatial Algorithms and Systems (TSAS)* 8.3 (2022), pp. 1–26 (Cited on page 1).

[ZWC+23] B. Zhou, Z. Wu, Z. Chen, X. Liu, and Q. Li. "Wi-Fi RTT/encoder/INS-based robot indoor localization using smartphones". In: *IEEE Transactions on Vehicular Technology* 72.5 (2023), pp. 6683–6694 (Cited on pages 24 sq.).

[ZWD23] S. Zhang, A. Wijesinghe, and Z. Ding. "RME-GAN: A Learning Framework for Radio Map Estimation based on Conditional Generative Adversarial Network". In: *IEEE Internet of Things Journal* (2023) (Cited on pages 18, 25).

[ZWZ+17] Z. Zhao, J. Wang, X. Zhao, C. Peng, Q. Guo, and B. Wu. "NaviLight: Indoor localization and navigation under arbitrary lights". In: *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE. 2017, pp. 1–9 (Cited on pages 6, 121).

[ZWZ+23] L. Zhang, S. Wu, T. Zhang, and Q. Zhang. "Learning to locate: Adaptive fingerprint-based localization with few-shot relation learning in dynamic indoor environments". In: *IEEE Transactions on Wireless Communications* 22.8 (2023), pp. 5253–5264 (Cited on pages 17, 25).

[ZWZ23] Y. Zou, W. Wu, and Z. Zhang. "Source localization based on hybrid AOA, TDOA, and RSS measurements". In: *IEEE Sensors Journal* 23.14 (2023), pp. 16293–16302 (Cited on page 3).

[ZXC+20] J. Zhang, W. Xiao, B. Coifman, and J. P. Mills. "Vehicle tracking and speed estimation from roadside lidar". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 13 (2020), pp. 5597–5608 (Cited on page 3).

[ZXL+19] J. Zhao, H. Xu, H. Liu, J. Wu, Y. Zheng, and D. Wu. "Detection and tracking of pedestrians and vehicles using roadside LiDAR sensors". In: *Transportation research part C: emerging technologies* 100 (2019), pp. 68–87 (Cited on page 3).

[ZXW+19] Y. Zhao, J. Xu, J. Wu, J. Hao, and H. Qian. "Enhancing camera-based multimodal indoor localization with device-free movement measurement using WiFi". In: *IEEE Internet of Things Journal* 7.2 (2019), pp. 1024–1038 (Cited on page 17).

[ZXZ+14] J. Y. Zhu, J. Xu, A. X. Zheng, J. He, C. Wu, and V. O. Li. "WIFI fingerprinting indoor localization system based on spatio-temporal (S-T) metrics". In: *Int'l Conf. Indoor Positioning and Indoor Navigation*. 2014, pp. 611–614 (Cited on page 21).

[ZZ17] S. Zhu and X. Zhang. "Enabling high-precision visible light localization in today's buildings". In: *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*. 2017, pp. 96–108 (Cited on pages 6, 121).

[ZZ18] C. Zhang and X. Zhang. "Visible light localization using conventional light fixtures and smartphones". In: *IEEE Transactions on Mobile Computing* 18.12 (2018), pp. 2968–2983 (Cited on pages 6, 121).

[ZZH+22] Y. Zhuang, C. Zhang, J. Huai, Y. Li, L. Chen, and R. Chen. "Bluetooth localization technology: Principles, applications, and future trends". In: *IEEE Internet of Things Journal* 9.23 (2022), pp. 23506–23524 (Cited on page 4).

[ZZL+23] X. Zhang, Y. Zhang, G. Liu, and T. Jiang. "Autoloc: Toward ubiquitous aoa-based indoor localization using commodity wifi". In: *IEEE Transactions on Vehicular Technology* 72.6 (2023), pp. 8049–8060 (Cited on page 24).

[ZZX+15] Q. Zhang, Z. Zhou, W. Xu, J. Qi, C. Guo, P. Yi, T. Zhu, and S. Xiao. "Fingerprint-free tracking with dynamic enhanced field division". In: *IEEE INFOCOM*. 2015, pp. 2785–2793 (Cited on page 22).